

**PERANCANGAN DAN IMPLEMENTASI *HEADLAMP* PADA
SEPEDA MOTOR SEBAGAI PENGATUR JARAK AMAN
PANCARAN CAHAYA DENGAN MENGGUNAKAN
SISTEM *LED MATRIX***

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh :

Allif Maulana

NIM : 125150307111022



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

**PERANCANGAN DAN IMPLEMENTASI HEADLAMP PADA SEPEDA MOTOR SEBAGAI
PENGATUR JARAK AMAN PANCARAN CAHAYA DENGAN MENGGUNAKAN
SISTEM LED MATRIX**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

**Disusun Oleh :
Allif Maulana
NIM: 125150307111022**

**Skripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2018
Telah diperiksa dan disetujui oleh:**

Dosen Pembimbing I

Dosen Pembimbing II

**Rizal Maulana, S.T, M.T., M.Sc.
NIK. 20160789 1009 001**

**Hurriyatul Fitriyah, S.T, M.Sc.
NIK. 19851001 201504 002**

**Mengetahui
Ketua Jurusan Teknik Informatika**

**Tri Astoto Kurniawan, S.T., M.T. Ph.D.
NIP: 197105182003121001**

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Januari 2018

Allif Maulana

NIM: 125150307111022

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah Swt. yang melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang menjadi salah satu syarat untuk menyelesaikan program studi Sistem Komputer jenjang Strata-1 Universitas Brawijaya Malang. Shalawat serta salam semoga tetap tercurah kepada penutup para nabi, Muhammad Shallallahu 'alaihi wa sallam, keluarga, sahabat dan para pengikutnya hingga hari kiamat.

Penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah banyak memberikan bantuan, masukan, nasehat, bimbingan dan dukungan. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang tak terhingga khususnya kepada :

1. Kedua orang tua tercinta Bapak dan Ibu yang selalu mendukung, mendoakan, memberi nasehat, petunjuk, dan selalu berkerlingat siang dan malam mencari uang demi penulis.
2. Bapak **Sabriansyah Rizkiqa Akbar, S.T., M.Eng** selaku Ketua Program Studi Teknik Komputer serta dosen PA yang sudah banyak membantu.
3. Bapak **Rizal Maulana, S.T, M.T., M.Sc.** selaku dosen Pembimbing I, yang telah memberi bimbingan, pengarahan-pengarahan dan masukan kepada penulis hingga terselesainya Skripsi ini.
4. Ibu **Hurriyatul Fitriyah, S.T, M.Sc.** selaku dosen Pembimbing II, selaku co-dosen pembimbing yang telah memberi bimbingan, pengarahan-pengarahan dan masukan kepada penulis hingga terselesainya Skripsi ini.
5. Segenap Bapak Ibu dosen dan staf serta karyawan Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan yang diberikan.

Penulis menyadari bahwa dalam penyusunan skripsi ini banyak kekurangannya. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat menyempurnakan penulisan ini sehingga dapat bermanfaat dan berguna untuk perkembangan ilmu pengetahuan. Amin.

Malang, 18 Januari 2018

Penulis

ABSTRAK

Pencahayaan pada kendaraan bermotor dengan sistem pengontrolan pancaran cahaya sangat dibutuhkan saat ini. Sehingga dibuat sebuah sistem headlamp yang mampu mengontrol pancaran cahaya tanpa perlu adanya pengontrolan dari pengguna kendaraan tersebut. Pembuatan sistem lampu ini menggunakan sebuah kamera *webcam* yang berguna untuk membaca kondisi jalan, lalu sebuah mikrokomputer yang digunakan untuk memproses hasil kondisi jalan tersebut dengan menggunakan metode *hough circle transform*. Dari hasil proses tersebut digunakan untuk mengkondisikan lampu *led matrix* dalam mengatur pancaran cahaya. Dari hasil pengujian didapatkan bahwa sistem ini mampu mendeteksi kendaraan dengan menggunakan pancaran cahaya sebagai acuan kendaraan lain. Pengontrolan lampu *led matrix* dapat bekerja dengan kondisi yang telah ditentukan. Dan hasil dari validitasi sistem menjelaskan bahwa *input* video dari kamera *webcam* dapat digunakan untuk mengontrol lampu *led matrix* dengan valid.

Kata kunci: *led matrix, hough circle tranform, pendeteksi kendaraan, headlamp*

ABSTRACT

Lighting on motor vehicles with light emission control systems is essential today. So made a headlamp system that is able to control the light emitted without the need for control of the user of the vehicle. Making this lamp system using a webcam camera that is useful to read the condition of the road, then a microcomputer that is used to process the results of the road condition by using the hough circle transform method. From the results of the process is used to condition the led matrix lamp in order to emit light. From the test results obtained that the system is able to detect the vehicle by using the light beam as a reference other vehicles. Control of led matrix lamps can work with predetermined conditions. And the result of system validation explains that video input from webcam camera can be used to control the led matrix lamp with valid.

Keywords: led matrix, hough circle tranform, vehicle detector, headlamp

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah	2
1.6 Sistematika Pembahasan	2
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Tinjauan Pustaka.....	4
2.2 Dasar teori	5
2.2.1 Prinsip <i>Matrix LED Headlights</i>	5
2.2.2 Metode Pemrosesan Citra dengan Pendeteksian Objek.....	6
2.2.3 Deteksi Warna RGB dan HSV	7
2.2.4 Thresholding	8
2.2.5 Smoothing.....	9
2.2.6 Operasi Deteksi Tepi <i>Canny</i>	10
2.2.7 Hough Circle Transform	11
2.2.8 Library OpenCV.....	13
2.2.9 <i>Library Wiring Pi</i>	13
2.2.10 <i>Embedded System</i>	15
2.2.11 Web Camera	16
2.2.12 Driver Penurun Tegangan	16
BAB 3 METODOLOGI	18
3.1 Metode Penelitian	18

3.1.1Kebutuhan Fungsional	18
3.1.2Kebutuhan Non – Fungsional	18
3.2 Desain Sistem	19
3.3 Rekayasa kebutuhan	19
3.3.1Kebutuhan Perangkat Keras	19
3.3.2Kebutuhan Perangkat Lunak.....	20
3.4 Perancangan Sistem.....	20
3.4.1Perancangan Desain <i>Headlamp</i>	20
3.4.2Perancangan Sumber Daya.....	20
3.4.3Perancangan Rangkaian Transistor	20
3.5 Implementasi Sistem.....	20
3.5.1Implementasi Pendeteksi Kendaraan	21
3.5.2Implementasi Desain <i>Headlamp</i>	21
3.5.3Implementasi Sistem Pada Sepeda Motor	21
3.6 Pengujian Dan Analisis	21
3.6.1Pengujian Pendeteksi Kendaraan	21
3.6.2Pengujian Pengontrolan Lampu	21
3.6.3Pengujian Penggabungan sistem.....	22
3.7Kesimpulan	22
BAB 4 REKAYASA KEBUTUHAN.....	23
4.1 Gambaran Umum Sistem.....	23
4.1.1 Tujuan	23
4.1.2 Kegunaan	23
4.1.3 Karakteristik Pengguna	23
4.1.4 Lingkungan Operasi	24
4.2 Kebutuhan Antarmuka Eksternal.....	24
4.2.1 Kebutuhan Antarmuka Pengguna.....	24
4.2.2 Kebutuhan Perangkat Keras	24
4.3 Kebutuhan Fungsional	24
4.3.1 Dapat Membedakan Pengendara Lain Dengan Jalan	24
4.3.2 Dapat Menerangi Jalan Lebih Baik Dari Sebelumnya	25
4.4 Kebutuhan Non Fungsional.....	25
4.4.1 Kebutuhan Performansi.....	25
4.4.2 Kebutuhan Keselamatan.....	25

4.5 Spesifikasi Perangkat Keras	25
4.5.1 Laptop	26
4.5.2 Mikrokomputer.....	26
4.5.3 WebcamLogitech C170	27
4.5.4 LedCreeX-TE	27
4.5.5 Stepdown XL4015	28
4.5.6 Transistor	28
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan Sistem.....	30
5.1.1 Gambaran Umum Sistem.....	30
5.1.2 Perancangan Perangkat Keras	31
5.1.3 Perancangan Perangkat Lunak	36
5.2 Implementasi Sistem	41
5.2.1 Implementasi Perangkat Keras	41
5.2.2 Implementasi Perangkat Lunak	44
BAB 6 PENGUJIAN DAN ANALISIS.....	51
6.1 Pengujian Pendeteksi Kendaraan	51
6.1.1 Tujuan	51
6.1.2 Prosedur Pengujian Pendeteksi Kendaraan.....	51
6.1.3 Hasil dari Pengujian	51
6.1.4 Analisis Pengujian	56
6.2 Pengujian Pengontrolan Lampu.....	56
6.2.1 Tujuan	56
6.2.2 Prosedur Pengujian Pendeteksi Kendaraan.....	56
6.2.3 Hasil dari Pengujian	56
6.2.4 Analisis Pengujian	60
6.3 Pengujian ValiditasSistem	60
6.3.1 Tujuan	60
6.3.2 Prosedur Pengujian ValiditasSistem	60
6.3.3 Hasil dari Pengujian	61
6.3.4 Analisis Pengujian	65
PENUTUP	66
DAFTAR PUSTAKA	67
LAMPIRAN	69

DAFTAR GAMBAR

Gambar 2.1	Prototype lampu headlamp	4
Gambar 2.2	Desain headlamp AUDI , Mercedes-Benz dan BMW	5
Gambar 2.3	Sistem kerja lampu <i>Matrix LED Headlights</i>	5
Gambar 2.4	Sistem lampu <i>Matrix LED Headlights</i> ketika mendeteksi cahaya	6
Gambar 2.5	Pengolahan sinyal pada Citra Digital	6
Gambar 2.6	Koordinat Citra Digital	7
Gambar 2.7	Kubus Model RGB	7
Gambar 2.8	Kerucut HSV	8
Gambar 2.9	Contoh hasil thresholding	9
Gambar 2.10	Hasil Gaussian Smoothing	9
Gambar 2.11	Hasil dari transformasi Gaussian Blur 1 Dimensi	10
Gambar 2.12	Hasil dari transformasi Gaussian Blur 2D	10
Gambar 2.13	Hasil deteksi tepian dengan (a) <i>threshold</i> tunggal (b) <i>threshold</i> tinggi dan rendah	11
Gambar 2.14	Hasil deteksi tepian <i>Canny</i>	11
Gambar 2.15	Proses mendeteksi lingkaran dengan Hough Circle Transform	12
Gambar 2.16	Hasil pendeteksian citra dengan Hough Circle Transform	12
Gambar 2.17	Tampilan Hasil Pembacaan GPIO	15
Gambar 2.18	<i>Raspberry Pi</i>	15
Gambar 2.19	Webcam	16
Gambar 2.20	<i>Stepdown</i> LM2596	17
Gambar 3.1	Flowchart Pengerjaan Penelitian	18
Gambar 3.2	Desain Sistem	19
Gambar 4.1	Laptop Thosiba L745	26
Gambar 4.2	<i>Raspberry Pi 3 Model B</i>	26
Gambar 4.3	Logitech C170	27
Gambar 4.4	<i>LedCreeX-TE</i> 5W	28
Gambar 4.5	<i>Stepdown</i> XL4015 DC – DC	28
Gambar 4.6	<i>Transistor</i> BD-139	29
Gambar 5.1	Gambaran Umum Sistem	30
Gambar 5.2	Desain pada sepeda motor	31
Gambar 5.3	Desain Akhir Lampu Utama	31
Gambar 5.4	Desain Peletakan Mikrokomputer	32

Gambar 5.5Desain Peletakan Rangkaian <i>Stepdown</i>	32
Gambar 5.6Desain Peletakan Kipas Pendingin	33
Gambar 5.7Desain Pelatakan <i>Led Matrix</i>	33
Gambar 5.8Desain Bagian Lampu	34
Gambar 5.9Skematic Rangkaian Alat	35
Gambar 5.10Susunan Led pada Headlamp.....	36
Gambar 5.11Diagram Alir Sistem.....	36
Gambar 5.12FlowchartInput Video.....	37
Gambar 5.13FlowchartPre-processing.....	38
Gambar 5.14FlowchartSmoothing	39
Gambar 5.15FlowchartHoughCircleTransform	40
Gambar 5.16PreudeoCode Pembuatan Garis	40
Gambar 5.17Implementasi Lampu Sepeda Motor	41
Gambar 5.18Implementasi Peletakan Kamera	42
Gambar 5.19Implementasi Desain <i>Headlamp</i>	42
Gambar 5.20Implementasi Mikrokomputer dan Rangkaian <i>Transistor</i>	43
Gambar 5.21Implementasi <i>Stepdown</i> (a) untuk <i>RaspberryPi</i> (b) untuk <i>Led</i>	43
Gambar 5.22Implementasi <i>Led</i>	43
Gambar 5.23Hasil Input Video	44
Gambar 5.24Hasil <i>Resize</i> ,(a) Sebelum <i>Resize</i> , (b) Sesudah <i>Resize</i>	45
Gambar 5.25Hasil Conversi Citra, (a) sebelum diconversi, (b) sesudah dikonversi.....	45
Gambar 5.26Hasil Thresholding.....	46
Gambar 5.27Hasil Smoothing	47
Gambar 5.28Hasil <i>Hough Circle Tranform</i> , (a) Frame Tanpa <i>Hough Circle Tranform</i> , (b) Frame Dengan <i>Hough Circle Tranform</i>	48
Gambar 6.1Gambaran Sederhana Pembacaan Area Berdasar Titik Kordinat	51
Gambar 6.2Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil <i>Thresholding</i>	52
Gambar 6.3Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil <i>Thresholding</i>	53
Gambar 6.4Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil <i>Thresholding</i>	53
Gambar 6.5Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil <i>Thresholding</i>	54
Gambar 6.6Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil <i>Thresholding</i>	55

Gambar 6.7Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil <i>Thresholding</i>	55
Gambar 6.8Kondisi percobaan lampu 1.....	57
Gambar 6.9Kondisi percobaan lampu 2.....	58
Gambar 6.10Kondisipercobaan lampu 3	59
Gambar 6.11Kondisi percobaan lampu 4.....	59
Gambar 6.12 Kondisi percobaan lampu 5.....	60

DAFTAR TABEL

Tabel 2.1Nilai sudut dan warna pada <i>Hue</i> (Hu, 2002)	8
Tabel 4.1Spesifikasi Laptop Thosiba L745.....	26
Tabel 4.2 Spesifikasi Mikrokomputer <i>Raspberry Pi 3</i>	27
Tabel 4.3Spesifikasi <i>Webcam Logitech C170</i>	27
Tabel 4.4Spesifikasi <i>Led Cree X-TE 5W</i>	28
Tabel 4.5Spesifikasi <i>Stepdown XL4015 DC – DC</i>	28
Tabel 4.6Spesifikasi <i>Transistor BD-139</i>	29
Tabel 5.1Kesimpulan jangkauan warna objek yang diinginkan	39
Tabel 5.2Potongan Program Input Video.....	44
Tabel 5.3Potongan Program <i>Pre-processing</i>	45
Tabel 5.4Potongan Program <i>Thresholding</i>	46
Tabel 5.5Potongan Program <i>Smoothing</i>	46
Tabel 5.6Potongan Program <i>HoughCircleTranform</i>	47
Tabel 5.7Potongan Program Titik Centroid.....	48
Tabel 5.8Potongan Program Pembuatan Garis Bantu	48
Tabel 5.9Potongan Program pengondisian <i>Led</i>	49
Tabel 6.1Potongan Program Deteksi Pancaran Cahaya dengan Area yang telah ditentukan	52
Tabel 6.2Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan.....	52
Tabel 6.3Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan.....	53
Tabel 6.4Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan.....	54
Tabel 6.5Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan.....	54
Tabel 6.6Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan.....	55
Tabel 6.7 Potongan Program Inisialisasi Pin GPIO <i>Raspberry Pi</i>	56
Tabel 6.8Potongan Program Pengaturan <i>Mode Pin</i>	56
Tabel 6.9Potongan Program Percobaan Pengondisian <i>Led1</i>	57
Tabel 6.10Potongan Program Percobaan Pengondisian <i>Led 2</i>	58

Tabel 6.11Potongan Program Percobaan Pengondisian <i>Led3</i>	58
Tabel 6.12Potongan Program Percobaan Pengondisian <i>Led4</i>	59
Tabel 6.13Potongan Program Percobaan Pengondisian <i>Led5</i>	59
Tabel 6.14 Validitasi Sistem Pendeteksi Objek dan Pengontrol Lampu.....	61

DAFTAR LAMPIRAN

L 1 Lampiran <i>Source Code</i>	69
---------------------------------------	----

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologisaat ini telah menghasilkan berbagai jenis sistem yang dapat membantu pengguna untuk mempermudah maupun meningkatkan keamanan dalam melakukan pekerjaan sehari – hari. Salahcontohnya teknologi pada *headlamp* kendaraan bermotor. Teknologi *headlamp* ini sistem menerangan jalan untuk kendaraan bermotor. Teknologi ini dapat digunakan untuk menerangi jalan maupun menerangilingkungan sekitar, sehingga pengguna kendaraan bermotor bisa lebih meningkatkan keamanan dalam berkendara.

Sistem penerangan pada kendaraan bermotor sekarang ini telah banyak menggunakan lampu *aftermarket* yang berbeda dengan lampu yang telah disediakan oleh pabrikan. Hal yang membedakan antara lampu dari pabrikan dengan lampu *aftermarket* terletak pada pencahayaannya. Pengendara akan merasakan adanya perubahan pencahayaan yang lebih terang jika menggunakan *aftermarket*. Namun penggunaan lampu *aftermarket* juga memiliki kelemahan jika pengaturan lampunya tidak sesuai, yaitu dapat menyebabkan pengendara lain menjadi silau karena terganggu oleh pancaran cahayanya yang dihasilkan. Hal ini sangatlah berbahaya, karena merupakan salah satu penyebab terjadinya kecelakaan. (Ipanase, 2014)

Lampu *aftermarket* memerlukan sebuah proyektor (tempat lampu) yang dapat mengatur pancaran cahaya lebih baik dari proyektor buatan pabrik. Pemasangan proyektor harus merombak secara signifikan pada konstruksi *headlamp* kendaraan sehingga dapat dipasang secara permanen. (Rahmatullah, 2016)

Led matrix adalah sebuah sistem lampu yang dapat menyala bergantian maupun bersamaan. Sistem ini telah diterapkan pada berbagai alat yang telah ada, seperti pada layar yang terletak di jalan – jalan sebagai media promosi suatu produk. Sistem ini menggunakan lampu yang disusun sejajar dan dihidupkan sebagian untuk dapat menghasilkan sebuah karakter yang dapat dipahami.

Berdasarkan latar belakang tersebut, penelitian ini merancang tentang perancangan dan implementasi *headlamp* pada sepeda motor sebagai pengatur jarak aman pancaran cahaya dengan menggunakan sistem *led matrix*. Dengan menggunakan sebuah kamera yang digunakan untuk mendeteksi adanya pengendara lain dan menjadikannya *actuator* untuk mikrokontroler. Mikrokontroler akan mengelola hasil kamera dalam mengondisikan jalan dan menentukan lokasi cahaya yang akan dipancarkan. Dari hasil yang telah dikondisikan oleh mikrokontroler, kemudian mikrokontroler menggerakkan lampu dengan sistem *led matrix*.

Penelitian skripsi ini adalah untuk merancang dan mengimplementasi sebuah lampu utama yang dapat mengatur pancaran cahaya agar tidak menyilaukan

pengendara lain sehingga dapat digunakan untuk perkembangan teknologi dibidang pencahayaan pada kendaraan bermotor.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan menjadi beberapa masalah sebagai berikut :

1. Bagaimana cara mendeteksi kendaraan dengan menggunakan kamera.
2. Bagaimana cara mengontrol saklar pada lampu *led matrix*.
3. Bagaimana akurasi pendeteksian beserta fungsional lampu *led matrix*.

1.3 Tujuan

Tujuan penelitian dapat di uraikan seperti di bawah ini :

1. Mengetahui kondisi jalan yang akan dilalui agar dapat membantu peneliti dalam membangun sistem.
2. Memberikan pengontrolan lampu secara otomatis agar pengendara tidak lagi mengontrol lampu secara analog.
3. Mengetahui akurasi pendeteksian beserta fungsional lampu led matrix ketika digunakan di jalan raya.

1.4 Manfaat

Manfaat dalam penelitian ini dengan melakukan percobaan dalam pembuatan lampu yang lebih baik dari lampu aftermarket. Pembuatan lampu ini di desain sedemikian rupa agar dapat bekerja tanpa merombak bentuk headlamp dan pancaran cahaya sama seperti lampu headlamp yang menggunakan proyektor tambahan untuk pengaturan pancaran cahayanya.

1.5 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal :

1. Sistem hanya dapat bekerja ketika malam hari dengan kondisi tidak hujan.
2. Sudut pandang kamera 65° dan dapat memproses gambar dengan jarak maksimal 20 m.
3. Resolusi video yang digunakan sebesar 320x240.

1.6 Sistematika Pembahasan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, sebagai berikut.

BAB I PENDAHULUAN

Menguraikan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan dari “Perancangan Dan Implementasi *Headlamp* Pada Sepeda Motor Sebagai Jarak Aman Pancaran Cahaya Dengan Menggunakan Sistem *Led Matrix*”.

BAB II LANDASAN KEPUSTAKAAN

Menguraikan kajian pustaka dan dasar teori yang mendasari sistem led matrix, sistem pengontrolan lampu yang sekarang ada.

BAB III METODOLOGI

Menguraikan tentang metode dan langkah kerja yang terdiri dari studi literatur, analisis kebutuhan simulasi, perancangan sistem, implementasi dan analisis serta pengambilan kesimpulan.

BAB IV REKAYASA KEBUTUHAN

Bab ini berisi tentang perancangan sistem lampu pada sepeda motor dengan menggunakan raspberry pi, meliputi perancangan hardware.

BAB V PERANCANGAN DAN IMPLEMENTASI

membahas tentang implementasi dari sistem lampu pada sepeda motor menggunakan raspberry pi.

BAB VI PENGUJIAN DAN ANALISIS

Membuat proses dan hasil pengujian terhadap sistem lampu pada sepeda motor dengan menggunakan raspberry pi yang telah direalisasikan.

BAB VII PENUTUP

bab ini menguraikan kesimpulan dari hasil perancangan sistem yang telah dibuat. Kemudian diberi saran dari hasil pengujian yang telah terjadi serta pengembangan dari sistem yang perlu ditingkatkan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tinjauan pustaka yang meliputi tinjauan pustaka dari dasar teori yang diperlukan untuk penelitian. Tinjauan pustaka membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Tinjauan Pustaka

Pada jurnal internasional dengan judul *"Programmable Automotive Headlights"* oleh Robert Tamburo pada tahun 2014, penelitian melakukan riset bagaimana merancang sebuah sistem lampu utama pada kendaraan peroda empat dengan sistem lcd proyektor. Sistem pendeteksi pada riset ini menggunakan kamera dengan metode image processing. Sistem ini memiliki kelemahan, yaitu perangkat keras yang digunakan cukup besar sehingga penggunaan ini tidak dapat di implementasikan pada sepeda motor, karena pada headlamp sepeda motor memiliki ukuran yang kecil. (Tamburo, 2014)



Gambar 2.1Prototype lampu headlamp

(Tamburo, 2014)

Pada jurnal internasional dengan judul *"A Comprehensive Survey Of Vision Based Vehicle Intellifent Front Light System"* oleh Feng Luo dan Fengjian HU pada tahun 2014. Peneliti melakukan riset penggunaan headlamp yang telah ada pada kendaraan ternama yaitu Audi, Mercedes-Benz, dan BMW. Sistem kerja teknologi ini berasal dari desain headlamp yang telah dibuat agar memaksimalkan kinerja dari sistem lampu tersebut. Sistem ini memiliki kelemahan, yaitu adanya perubahan pada headlamp sepeda motor jika diimplementasikan. (Feng, 2014)



Gambar 2.2Desain headlamp AUDI , Mercedes-Benz dan BMW

(Feng, 2014)

Dari kedua jurnal yang telah di jelaskan, tidak dapat digunakan di sepeda motor. Karena itu perlu adanya sebuah rancangan yang dapat dimplementasikan pada sepeda motor dengan sistem yang sama tanpa adanya perombakan secara signifikan pada kontruksi *headlamp* kendaraan tersebut.

2.2 Dasar teori

Dasar teori membahas tentang teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.2.1 Prinsip *Matrix LED Headlights*

Matrix LED Headlights adalah suatu teknologi dengan menggunakan 25 segmen yang dikontrol secara individu per lampu yang memungkinkan 966.105.422 variasi pancaran cahaya yang berbeda. Sistem ini bekerja dengan sebuah kamera yang mendeteksi sumber cahaya dari kendaraan yang melaju dengan *real time*.(Fallah, 2013)



Gambar 2.3 Sistem kerja lampu *Matrix LED Headlights*

(Fallah, 2013)

Sistem ini bekerja dengan mendeteksi lampu depan atau lampu belakang kendaraan dan mengirimkan informasi ini ke unit kontrol. Kemudian unit kontrol akan memanipulasi 25 segmen untuk meredupkan zona pencahayaan beberapa segmen yang mempengaruhi pengendara lain sebelum pengendara lain mendekat. Sistem ini akan mematikan beberapa segmen zona pencahayaan untuk menjaga pengendara lain tidak silau dan sisa segmen zona pencahayaan yang lainnya tetap menyala untuk menerangi jalan dan pinggir jalan.(Spinks, 2016)

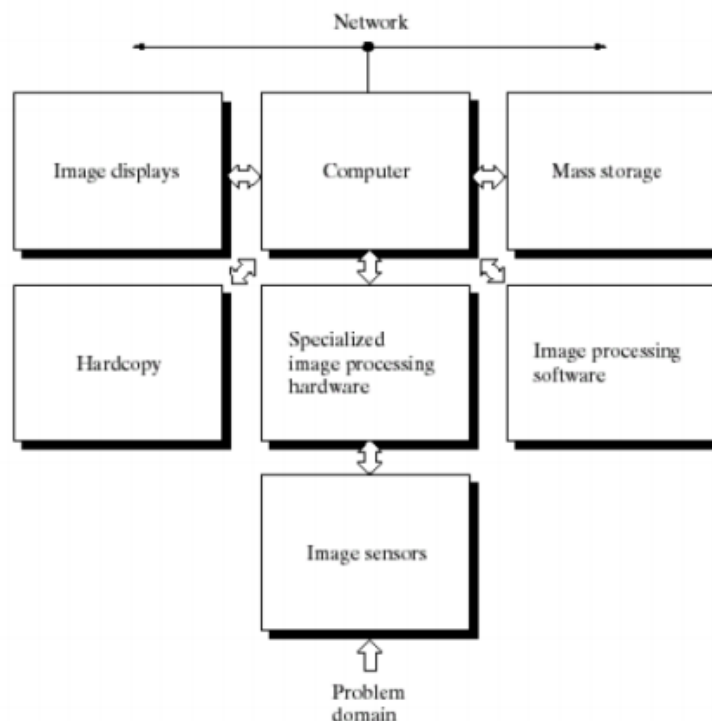


Gambar 2.4 Sistem lampu *Matrix LED Headlights* ketika mendeteksi cahaya

(Spinks, 2016)

2.2.2 Metode Pemrosesan Citra dengan Pendeteksian Objek

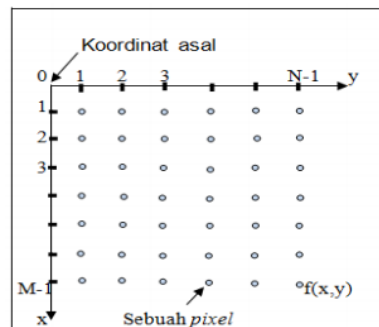
Pada Gambar 2.5, dapat diketahui, pengambilan citra secara fisik, dilakukan oleh sebuah sensor yang peka terhadap radiasi dari objek. Dari hasil tangkapan tersebut, kemudian dikonversi menjadi biner oleh *digitizer / hardware* tambahan yang dapat mengolah menjadi data *digital*. Pada penelitian ini, tidak ada perangkat keras tambahan yang digunakan karena penggunaan masih sederhana. (Hu, 2002)



Gambar 2.5 Pengolahan sinyal pada Citra Digital

(Hu, 2002)

Citra digital adalah kumpulan nilai dengan deretan bit tertentu dalam membentuk sebuah *array*. Citra dapat didefinisikan sebagai fungsi $f(x,y)$ yang memiliki ukuran M baris dan N kolom dimana x dan y merupakan kordinat spasial. Amplitudof pada titik (x,y) sebagai intensitas atau tingkatan keabuan dari citra pada titik tersebut. Suatu citra dapat dikatakan sebagai citra digital apabila nilai x,y dan nilai amplitudo f keseluruhan memiliki nilai berhingga (finite) dan bernilai diskrit.(Putra, 2010)

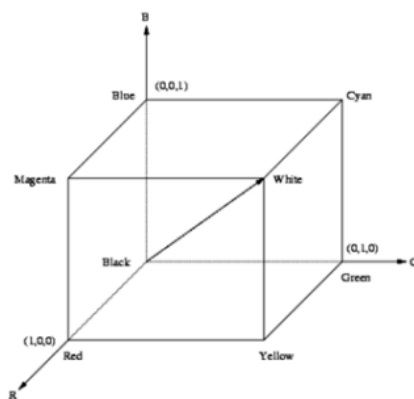


Gambar 2.6Koordinat Citra Digital

(Putra, 2010)

2.2.3 Deteksi Warna RGB dan HSV

RGB (*Red, Green, Blue*) adalah citra pada gambar terdiri dari tiga pola citra independen, satu di masing – masing warna primer: merah, hijau dan biru. Komposisi warna tertentu adalah dengan menentukan jumlah dari masing – masing komponen warna utama. Pada Gambar 2.7 ditunjukkan tentang geometri dari model warna RGB untuk menentukan warna dengan menggunakan sistem kordinat *Cartesian*. Spektrum *Greyscale*, yaitu warna-warna yang terbuat dari jumlah yang sama dari masing-masing primer, terletak pada garis yang menghubungkan simpul hitam dan putih.



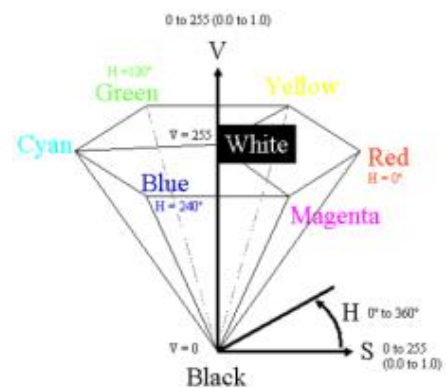
Gambar 2.7Kubus Model RGB

(Hu, 2002)

Misalkan saja warna kuning, ia adalah hasil (merah + hijau), *cyan* (biru +hijau) ,*magenta* (merah +biru), dan putih (merah+ hijau+biru). Sedangkan modelHSV yang terdiri dari 3 komponen yaitu *hue*, *saturation*, *value*. *Value* terkadangdisamakan dengan tingkat kecerahan. Model HSV dibuat oleh Alvy Ray Smithpada tahun 1978. HSV dikenal juga sebagai model warna *hex-cone*. *Hue*, merepresentasikan warna, hue adalah sudut diantara 0 sampai dengan 360 derajat.Konsep model warna HSV ditunjukkan oleh Tabel 2.1 tentang konsep sudut dan warna pada derajat Hue, dan Gambar 2.8 tentang Diagram model warna HSV dalam bentuk kerucut.

Tabel 2.1Nilai sudut dan warna pada *Hue*(Hu, 2002)

Sudut (°)	Warna
0 – 60	Merah
60 – 120	Kuning
120 – 180	Hijau
180 – 240	Cyan
240 – 300	Biru
300 – 360	Magenta



Gambar 2.8Kerucut HSV

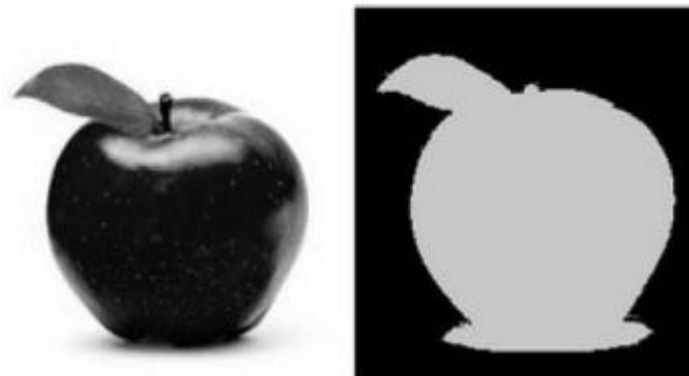
(Hu, 2002)

Pada Gambar 2.8 dijelaskan konsep *Saturation* mengindikasikan *range*/batas dari tingkat keabu-abuan dalam ruang warna/*color space*. Yaitu dari nilai 0-255(*Grayscale*), Jadi ada keadaan abu-abu antara nilai 0 dan 1. *Saturation* akan jadi ukuran operasi *threshold* pada penelitian ini. *Value* adalah tingkat kecerahan (*brightness*) dari warna dan nilainya bervariasi dan berhubungan dengan *saturation*. Nilainya berkisar dari 0-100%. Saat 0 nilainya hitam total, dengan peningkatan nilai *value* maka tingkat kecerahannya bertambah. Kelebihan model HSV adalah karena mendekati warna yang dirasa indra manusia, sehingga sering digunakan dalam pengolahan citra terutama pendeteksian suatu objek dengan warna tertentu.

2.2.4 Thresholding

Thresholding merupakan konversi citra berwarna ke citra biner yang dilakukan dengan cara mengelompokkan nilai derajat keabuan setiap *pixel* kedalam 2 kelas, hitam dan putih. Pada citra hitam putih terdapat 256 level, artinya mempunyai skala “0” sampai “255” atau [0, 255], dalam hal ini nilai intensitas 0 menyatakan hitam, dan nilai intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih. Kemudian dapat dipisahkan objek menurut rentan nilai

thresholding tersebut, sedang nilai *thresholding* lain diabaikan. (Kaehler, 2008). Contoh hasil *thresholding* adalah pada Gambar 2.9.

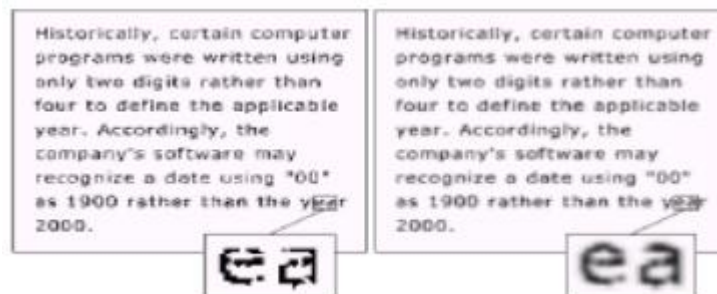


Gambar 2.9 Contoh hasil thresholding

(Kaehler, 2008)

2.2.5 Smoothing

Smoothing atau sering disebut *blurring*, adalah metode untuk mengkaburkan objek. Tujuannya adalah untuk mengurangi *noise*, sehingga mempermudah pemrosesan citra, terutama proses pendeteksian tepi. Hal ini karena proses *smoothing* dapat menghaluskan/ menyatukan bagian citra yang terpisah. (Hu, 2002) Hal ini terlihat seperti pada Gambar 2.10.



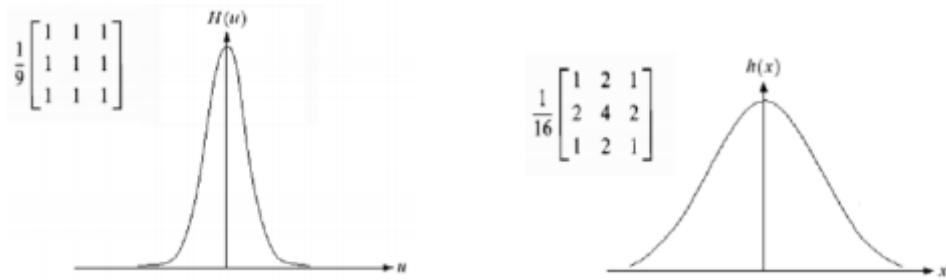
Gambar 2.10 Hasil Gaussian Smoothing

(Hu, 2002)

Filter yang digunakan untuk *smoothing* adalah *linear*, dengan 1 dimensi. Persamaan untuk *smoothing* 1 dimensi dapat dilihat pada Persamaan (2.1).

$$\begin{cases} H(u) = e^{-\frac{u^2}{2\sigma^2}} \\ h(x) = \sqrt{2\pi} \cdot A e^{-\frac{1}{2}\pi^2 \sigma^2 x^2} \end{cases} \quad (2.1)$$

Pada Persamaan (2.1) terdapat persamaan $H(u)$ yang berupa persamaan citra, dan persamaan $h(x)$ yang berupa hasil *smoothing* dengan 1 Dimensi. Terlihat bahwa hasil *smoothing* adalah seperti *low pass filter*. (Hu, 2002) Grafik median dari kedua persamaan ini terlihat pada Gambar 2.11.



(a) Hasil dari transformasi $H(u)$, citra awal (b) Hasil setelah do smoothing, $h(x)$

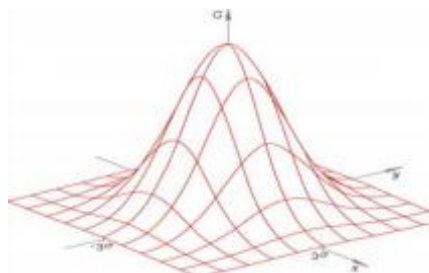
Gambar 2.11 Hasil dari transformasi Gaussian Blur 1 Dimensi

(Hu, 2002)

Pada persamaan *smoothing* yang digunakan pada penelitian ini adalah 2 dimensi . Persamaan ini tertulis pada Persamaan (2.2 a). Nilai σ , pada OpenCV berasal dari Persamaan (2.2 b), n adalah parameter 1 untuk kernel horizontal atau parameter 2 untuk kernel vertikal. Kernel adalah matriks yang digunakan untuk proses konvolusi (OpenCV, 2012). Hasil grafik median dari persamaan *smoothing* 2D, dicontohkan pada Gambar 2.12.

$$G(x, y) = 2\pi \cdot A e^{-2\pi^2 \sigma^2 (x^2 + y^2)} \quad (2.2 a)$$

$$\sigma = 0,3 \left(\frac{n}{2} - 1 \right) + 0,8, n \quad (2.2 b)$$



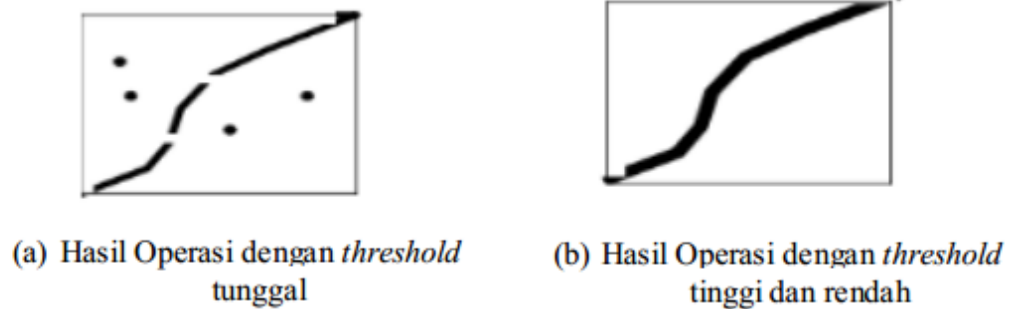
Gambar 2.12 Hasil dari transformasi Gaussian Blur 2D

(Hu, 2002)

2.2.6 Operasi Deteksi Tepi *Canny*

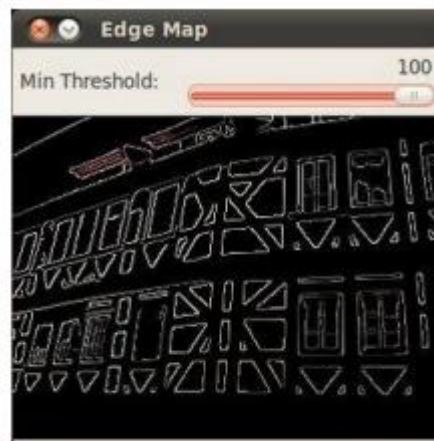
Operasi *CannyThreshold* didapatkan nilai tepian dari suatu objek. Dengan kata lain operasi *canny* akan membinnerisasikan nilai *threshold* yang dianggap tepian objek. Metode pada *Canny* adalah dengan *Hysteresis Threshold*. Dengan proses tersebut deteksi tepi adalah pixel tingkat keabuan lebih dari atau sama dengan *threshold* atas, sedangkan pixel dengan tingkat keabuan lebih dari *threshold* bawah, yang sama terhubung dengan pixel yang dianggap tepian objek

(lebih besar dari *threshold* atas) maka akan dianggap sebagai tepian objek. Hal ini untuk menghubungkan tepian objek yang mungkin tidak memiliki gradasi yang sama tinggi dengan tepian lain. Ilustrasi deteksi dengan dua nilai *threshold* dan satu nilai *threshold* ditunjukkan pada gambar 2.13. Contoh hasil deteksi tepian *Canny* pada citra ditunjukkan pada gambar 2.14.



Gambar 2.13 Hasil deteksi tepian dengan (a) *threshold* tunggal (b) *threshold* tinggi dan rendah

(Hu, 2002)

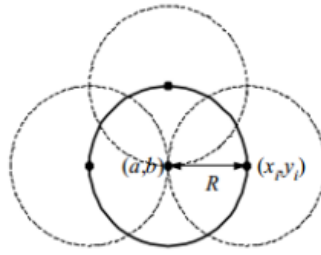


Gambar 2.14 Hasil deteksi tepian *Canny*

(Hu, 2002)

2.2.7 Hough Circle Transform

Persamaan *Hough* adalah salah satu persamaan untuk merepresentasikan Kontur (*countour*). Kontur adalah salah satu bentuk deteksi tepi, deteksi ini bisa merepresentasikan batas dari suatu daerah (*boundary region*) atau suatu bentuk. Maka dari itu didapatkan persamaan-persamaan untuk dapat merepresentasikan bentuk tersebut. (Munir, 2004) Salah satunya adalah *Hough Circle Detection*.



Gambar 2.15Proses mendeteksi lingkaran dengan Hough Circle Transform

(Munir, 2004)

Pada Gambar 2.13 dijelaskan tentang proses mendeteksi lingkaran dengan metode *Hough Circle Transform*. Ciri-ciri dari lingkaran adalah mempunyai titik tengah (a,b) , dengan tepian dengan jarak jari-jari yang sama. Prosesnya adalah mencari bentuk lingkaran di semua *pixel*. Maka dari itu metode yang digunakan adalah persamaan *polar*, yang dijelaskan pada Persamaan (2.3). Pada Persamaan (3.5) dijelaskan tentang bagaimana melakukan perhitungan untuk dapat mencari titik a , b , dan r (*jari-jari*). Persamaannya adalah sebagai berikut :

$$(x_i - a)^2 + (y_i - b)^2 = r^2 \quad (2.3 a)$$

$$x_i = a + r \cos \theta ; a = x_i - r \cos \theta \quad (2.3 b)$$

$$y_i = b + r \sin \theta ; b = y_i - r \sin \theta \quad (2.3 c)$$

$$r = \frac{x_i - a}{\cos \theta} ; r = \frac{y_i - b}{\sin \theta} \quad (2.3 d)$$

Transformasi lingkaran ditunjukkan pada Persamaan(2.3 a), kemudian diubah dalam bentuk perkalian sudut untuk dapat mencari jari-jari berdasarkan titik tengah (a,b) dan tepian (x,y) , perkalian sudut ini ditunjukkan pada Persamaan (2.3 b) dan Persamaan (2.3 c). Maka untuk mencari besar jari-jari lingkaran dapat diperoleh pada bentuk Persamaan (2.3 d).

OpenCV memiliki *library* untuk deteksi *Hough Transform*. Metode yang digunakan sama dengan *Hough Circle Transform* sederhana, hanya metodenya sudah dioptimasi, dengan tidak menghilangkan ciri *Hough Transform* yakni deteksi koordinat titik tengah (a,b) serta jari-jari lingkaran (r) .(OpenCV, 2012) Ilustrasi deteksi lingkaran ditunjukkan pada Gambar 2.14.



Gambar 2.16Hasil pendeteksian citra dengan Hough Circle Transform

(OpenCV, 2012)

2.2.8 Library OpenCV

OpenCV (*Open Source Computer Vision*) adalah *library* / pustaka fungsi pemrograman untuk pemrosesan waktu nyata pada *computer vision*. *OpenCV* dirilis dibawah lisensi *BSD*, gratis untuk digunakan untuk kegunaan akademis maupun untuk fungsi komersial. Menggunakan antarmuka *C++*, *C* dan *Python*, juga *Java*. OpenCV adalah *library Open Source* untuk *Computer Vision* untuk *C/C++*, *OpenCV* didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video* (Huaman, 2012). *Library* ini dapat berjalan pada sistem operasi *Windows*, *Linux* maupun *Mac*.

Feature yang dimiliki OpenCV antara lain :

- a) Citra dan video I/O (file dan kamera based input, *image/video* file output).
- b) Manipulasi Matriks dan Vektor beserta rutin-rutin aljabar linear (*products, solvers, eigenvalues, SVD*).
- c) Pemroses Citra fundamental (*filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids*).
- d) Kalibrasi kamera (*calibration patterns, estimasi fundamental matrix, estimasi homography, stereo correspondence*).
- e) Analisis gerakan (*optical flow, segmentation, tracking*).
- f) Graphical User Interface (*display image/video, penanganan keyboard dan mouse handling, scroll-bars*).
- g) Pelabelan citra (*line, conic, polygon, text drawing*).

OpenCV terdiri dari 3 library, yaitu:

- a) CV : untuk algoritma *Image processing* dan *Vision*.
- b) Highgui : untuk *GUI, Image* dan *Video I/O*.
- c) CXCore : Untuk struktur data, support XML dan fungsi-fungsi grafis.

2.2.9 Library Wiring Pi

Wiring Pi adalah sebuah *library* yang digunakan untuk mengaktifkan *pin* pada *raspberrypi*. Penggunaan *library* ini dijalankan dengan menggunakan sistem *root*, sehingga ketika memanggil program perlu adanya pemanggilan "*sudo*" sebelum nama program yang akan dijalankan. *Library Wiring Pi* dapat dipanggil tanpa harus menggunakan *root*, dengan cara menginisialisasi terlebih dahulu *pin* yang akan digunakan dengan sintax *system("gpio export 21 out")*, lalu panggil sintax *wiringPiSetupSys()* untuk dapat menjalankan *Wiring Pi*.

Untuk meng install wiringpi, pertama periksa apakah wiringPi belum terpasang. Di terminal, jalankan:

```
$ gpio -v
```

Jika Anda mendapatkan sesuatu, maka Anda sudah memasangnya. Langkah selanjutnya adalah mencari tahu apakah sudah terpasang melalui paket standar atau dari sumber. Jika Anda menginstalnya dari sumber, Anda tahu apa yang Anda lakukan - teruskan - tapi jika diinstal sebagai paket, Anda perlu menghapus paket terlebih dahulu. Untuk melakukan ini:

```
$ sudo apt-get purge wiringpi
```

```
$ hash -r
```

Jika Anda tidak menginstal GIT, maka di bawah salah satu rilis Debian (misalnya Raspbian), Anda dapat menginstalnya dengan:

```
$ sudo apt-get install git-core
```

Jika Anda mendapatkan kesalahan di sini, pastikan Pi Anda up to date dengan versi terbaru Raspbian: (ini adalah ide bagus untuk melakukan secara teratur)

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Untuk mendapatkan WiringPi menggunakan GIT:

```
$ cd
```

```
$ git clone git://git.drogon.net/wiringPi
```

Jika Anda sudah menggunakan operasi kloning untuk pertama kalinya, maka

```
$ cd ~/wiringPi
```

```
$ git pull origin
```

Akan mengambil versi update maka anda bisa menjalankan kembali script build di bawah ini.

Untuk membangun / install ada script baru yang disederhanakan:

```
$ cd ~/wiringPi
```

```
$ ./build
```

Skrip build yang baru akan mengkompilasi dan menginstalnya untuk Anda menggunakan perintah sudo pada satu titik, jadi Anda mungkin ingin memeriksa skrip sebelum menjalankannya.

```
pi@raspberrypi /var/www/mypihc/bin/dht22 $ gpio readall
```

Model B2											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	IN	1	3	4		5V			
3	9	SCL.1	IN	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT0	TxD	15	14
		0v			9	10	1	ALT0	RxD	16	15
17	0	GPIO. 0	OUT	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v			
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	0	IN	CE1	11	7
28	17	GPIO.17	IN	0	51	52	0	IN	GPIO.18	18	29
30	19	GPIO.19	IN	0	53	54	0	IN	GPIO.20	20	31
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
Model B2											

Gambar 2.17 Tampilan Hasil Pembacaan GPIO

Pada gambar 2.17 menjelaskan pin yang *hardware* yang digunakan, dimana disana tertulis bahwa terdapat nama *pin* dan nama *wiringPi* dengan *mode pin* yang terdaftar.

2.2.10 Embedded System

Embedded system adalah sebuah system yang ditanamkan pada suatu perangkat komputer dengan desain khusus untuk melakukan tugas yang spesifik (Azmi, 2006). *Arduino* dan *Raspberry Pi* merupakan *platform* yang sangat sering digunakan dalam pengembangan *Embedded System*. *Raspberry Pi* adalah sebuah komputer berbentuk kecil dengan dimensi: 85.60 mm x 53.98 mm x 17 mm, dan memiliki berat hanya 45 g. *Board* yang ada pada *Raspberry Pi* memiliki sebuah prosesor serta grafik *chip* memori RAM dengan berbagai *interface* dan konektor yang mendukung perangkat eksternal (Vujovic, 2015). Berikut ini contoh *Raspberry Pi* yang ditunjukkan pada gambar 2.15.



Gambar 2.18 Raspberry Pi

Sumber: Raspberry, 2015

Seperti halnya dengan komputer yang lain, *Raspberry Pi* juga menggunakan sistem operasi bernama *Raspbian* dengan basis *Linux*. *Linux* merupakan sistem

operasi yang tepat bagi *Raspberry Pi* karena *Linux* bersifat *open source* dan gratis.(Vujovic, 2015)

2.2.11 Web Camera

Web camera atau yang biasa disebut dengan *webcam* merupakan kamera yang hasil tangkapannya bisa dilihat menggunakan aplikasi pada komputer. *Webcam* dapat didefinisikan sebagai kamera digital yang memiliki bentuk kecil serta resolusi yang rendah. Sebagian *webcam* juga difasilitasi dengan aplikasi yang dapat mendeteksi suara dan gerakan objek sehingga sering digunakan untuk sistem keamanan. Gambar atau video yang dihasilkan oleh webcam dapat disimpan dalam komputer atau diunggah ke internet(Wibowo, 2010). Berikut ini contoh webcam yang ditunjukkan pada gambar 2.19.



Gambar 2.19Webcam

(Logitech, 2015)

2.2.12 Driver Penurun Tegangan

Untuk menurunkan tegangan *voltase* dengan *input fix* atau tetap. Misalnya menurunkan 5VDC ke 3 VDC. Menggunakan *potensio* / *resistor* untuk menurunkan tegangan lebih mudah dan murah untuk dibuat, hanya saja dipengaruhi tegangan *input* DC ke *output* DC. Bila sumber *voltase* DC turun, maka *output* dari tahanan ikut turun. Kecuali sumber DC *input* adalah tetap. Seperti dari adaptor DC. (Obengplus, 2015)

Kendala kedua untuk menurunkan tegangan dengan tahanan seperti *resistor* atau *potensio* belum tentu bisa menurunkan *current* besar (*Ampere* besar seperti 0,5*Ampere* atau 1 *Ampere*). Seperti dalam 1 *ampere* secara perlahan *resistor* menjadi panas dan belahan akan merusak tahanan / *potensio*, kecuali menggunakan tipe komponen *resistor* khusus yang mampu dilewati beban *ampere* tinggi.(Obengplus, 2015)



Gambar 2.20 Stepdown LM2596

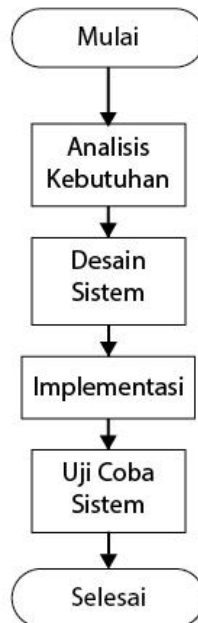
(Obengplus, 2015)

Salah satu sirkuit penurun tegangan dengan tipe LM2596 memiliki keunikan yaitu memiliki *output* stabil sesuai kebutuhan tanpa menentukan *input* yang dibutuhkan.

BAB 3 METODOLOGI

Pada bab ini menjelaskan metode yang akan digunakan untuk penelitian. Selain hal tersebut dalam bab ini akan menjabarkan tujuan dan cara dari setiap langkah yang akan dilakukan dalam penelitian.

3.1 Metode Penelitian



Gambar 3.1 Flowchart Pengerjaan Penelitian

Pada tahap ini merupakan tahap dalam menganalisa kebutuhan sistem yang akan digunakan. Sistem ini menggunakan dua komponen yaitu komponen hardware dan komponen software. Kebutuhan ini dapat digolongkan menjadi dua kebutuhan, yaitu kebutuhan fungsional dan non – fungsional

3.1.1 Kebutuhan Fungsional

Untuk kebutuhan fungsional adalah sebagai berikut :

- Sistem dapat mematikan sebagian pencahayaan ketika terdeteksi pengendara lain.
- Sistem dapat bekerja pada sebuah sepeda motor yang digunakan dalam penelitian ini.

Pada penjelasan kebutuhan fungsional yang telah dijelaskan diatas bertujuan untuk menjelaskan sistem yang akan ada pada penelitian ini.

3.1.2 Kebutuhan Non – Fungsional

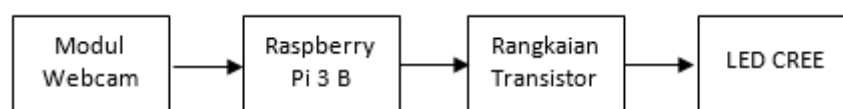
Untuk kebutuhan non – fungsional adalah sebagai berikut :

- *Raspberry pi 3* sebagai penerima hasil kamera dan pengontrol lampu yang akan digunakan untuk penerangan.
- Wifi atau Lan untuk media komunikasi antara *raspberry pi* dengan laptop dalam memprogram sistem yang akan digunakan pada penelitian ini.
- *Remote Desktop Connection* adalah program bawaan windows untuk menampilkan desktop *raspberry pi*.

Pada penjelasan kebutuhan non – fungsional yang telah dijelaskan diatas bertujuan untuk menjelaskan kebutuhan untuk memprogram *raspberry pi* dengan menggunakan laptop.

3.2 Desain Sistem

Desain sistem pada Gambar 3.2 ini merupakan alur perangkat keras yang digunakan dalam penelitian. Modul *webcam* digunakan sebagai modul untuk mendeteksi kendaraan lain. *Raspberry Pi 3 B* digunakan untuk memproses video hasil *webcam* untuk mengontrol lampu. Rangkaian transistor digunakan untuk mengontrol lampu dengan sumber listrik dari sepeda motor. Lampu *LEDCREE* digunakan sebagai penerangan jalan sepeda motor.



Gambar 3.2 Desain Sistem

3.3 Rekayasa kebutuhan

Analisa kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dalam merancang sistem yang akan di bangun dan di uji. Analisa kebutuhan dilakukan dengan cara menentukan kebutuhan dari sistem dan peralatan yang digunakan. Dalam kebutuhan sistem, terjadi proses komunikasi antar perangkat yang digunakan seperti perangkat keras dan perangkat lunak. Dengan adanya komunikasi tersebut dapat mempermudah untuk pembuatan sistem.

3.3.1 Kebutuhan Perangkat Keras

Sistem memerlukan beberapa perangkat keras untuk dapat memaksimalkan sistem secara keseluruhan. Raspberry Pi untuk pengontrol sistem yang mendapatkan input dari modul *webcam*. Modul *webcam* digunakan sebagai pendeteksi adanya pengendara lain disekitar pengguna. *Output* dari sistem ini adalah cahaya lampu dari sebuah kendaraan bermotor. Sebelum *raspberrypi* mengirim data ke lampu, terdapat rangkaian transistordigunakan untuk mematikan dan menghidupkan lampu.

3.3.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun sistem pengontrol lampu headlamp menggunakan *OS Rasbian* sebagai sistem operasi *raspberrypi* dan metode *imageprocessing* sebagai sistem utama pendeteksi pengendara lain dengan menggunakan *webcam*.

3.4 Perancangan Sistem

Pada perancangan sistem menjelaskan beberapa perancangan yang akan digunakan untuk membantu dalam pengerjaan penelitian ini.

3.4.1 Perancangan Desain Headlamp

Perancangan yang akan dilakukan untuk *headlamp* bertujuan untuk memaksimalkan sistem kerja lampu led. Karena lampu *led* memiliki sistem pencahayaan menyebar, sehingga perlu adanya desain *headlamp* untuk mengatur pancaran cahaya dari lampu *led* tersebut.

3.4.2 Perancangan Sumber Daya

Perancangan dalam sumber daya untuk sistem ini perlu diperhitungkan dikarenakan sistem ini akan bekerja pada sepeda motor. Sistem ini hanya memerlukan 3,3volt untuk lampu led dan 5volt untuk raspberry pi, sedangkan sumber daya yang dikeluarkan dari sepeda motor sekitar 12 volt hingga 14 volt ketika mesin hidup.

3.4.3 Perancangan Rangkaian Transistor

Perancangan pada rangkaian transistor ini ditujukan untuk sistem saklar pada lampu led yang akan dijalankan secara individu. Karena jika lampu led langsung mendapatkan sumber daya dari raspberry pi, lampu tidak akan bekerja secara maksimal disebabkan arus maksimal raspberry pi sekitar 2 ampere, dan untuk lampu sendiri akan bekerja secara maksimal ketika di aliri listrik dengan arus sebesar 1,5 ampere per lampu.

3.5 Implementasi Sistem

Implementasi sistem dilakukan dengan mengacu kepada perancangan hardware. Implementasi ini akan disusun sesuai dengan desain sistem. Komponen yang akan digunakan untuk hardware adalah raspberry pi, modul webcam, lampu led, stepdown, dan rangkaian transistor. Untuk implementasi software terletak pada pemrograman pengelola citra pada raspberry pi dengan menggunakan bahasa C++.

3.5.1 Implementasi Pendeteksi Kendaraan

Implementasi pendeteksi kendaraan dilakukan dengan menggunakan raspberry pi yang mendapatkan input berupa video dari sebuah module webcam. Raspberry pi memproses hasil gambar dengan menggunakan sistem opencv dengan metode Hough Circle Transform.

3.5.2 Implementasi Desain *Headlamp*

Implementasi ini dirancang untuk menentukan peletakan hardware yang akan digunakan pada sebuah headlamp sepeda motor. Dengan ukuran *headlamp* tersebut, sebisa mungkin hardware yang digunakan akan diletakkan didalam *headlamp*, sehingga tidak akan mengganggu sistem sepeda motor itu sendiri.

3.5.3 Implementasi Sistem Pada Sepeda Motor

Implementasi sistem pada sepeda motor dimana sistem akan diimplementasikan di sepeda motor untuk mengetahui kinerja sistem ketika berada di sepeda motor.

3.6 Pengujian Dan Analisis

Pengujian dan analisis yang akan dilakukan untuk mengetahui fungsionalitas sistem yang telah dirancang dan diimplementasikan. Pada penelitian ini terdapat beberapa pengujian yang akan dilakukan.

3.6.1 Pengujian Pendeteksi Kendaraan

3.6.1.1 Tujuan Pengujian

Tujuan dari pengujian pendeteksi kendaraan yaitu, untuk memastikan kendaraan terdeteksi ketika berpapasan di jalan. Apabila tidak dapat mendeteksi, maka sistem tidak akan jalan dengan semestinya.

3.6.1.2 Prosedur Pengujian

Pengujian dilakukan dengan cara menjalankan program *opencv* pada *raspberrypi* dan dihadapkan ke suatu objek yang menghasilkan cahaya. Sistem akan mendeteksi cahaya dengan cara ditampilkan pada monitor laptop dengan ditandai lingkaran pada objek tersebut. Jika tidak ada lingkaran pada monitor, maka sistem tidak dapat mendeteksi cahaya tersebut.

3.6.2 Pengujian Pengontrolan Lampu

3.6.2.1 Tujuan Pengujian

Tujuan dari pengujian pengontrolan lampu adalah untuk mengetahui lampu bekerja secara optimal ketika dihidupkan secara individu. Sehingga dari hasil pengontrolan lampu dapat diketahui lokasi hasil pancaran cahaya tiap lampunya.

3.6.2.2 Prosedur Pengujian

Pengujian dilakukan dengan cara menghidupkan tiap – tiap lampu bergantian dan dihadapkan pada dinding. Dari hasil pancaran cahaya tersebut, dapat ditentukan lokasi pancaran dan dilokasikan tiap – tiap lampu dengan melihat hasil dari kamera.

3.6.3 Pengujian Penggabungan sistem

3.6.3.1 Tujuan Pengujian

Tujuan dari pengujian penggabungan sistem ini untuk mengetahui sistem ini bekerja dengan *inputwebcam* dan output lampu. Dari hasil pengujian dapat diketahui masalah yang terjadi.

3.6.3.2 Prosedur Pengujian

Pengujian ini dilakukan dengan berada didepan alat dengan membawa sebuah senter sebagai pengendara lain. Ketika berada didepan alat, pancaran sinar masih mengganggu atau tidak. Jika masih, maka perlu diperbesar lokasi pancaran cahaya lampu. Sehingga ketika berada didepan alat, lampu akan mati sebelum pengendara lain berada di lokasi pancaaran cahaya lampu.

3.7 Kesimpulan

Pengambilan kesimpulan dapat dilakukan setelah tahapan perancangan implementasi dan pengujian telah usai. Kesimpulan yang di ambil untuk menjawab rumusan masalah yang sudah dijabarkan sebelumnya, tahapan terakhir dari penulis adalah sebuah saran untuk memperbaiki kesalahan yang terjadi pada implementasi lampu *headlamp* dan menyempurnakan penulisan serta mempertimbangkan atas hasil yang telah dilakukan.

BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Pada gambaran umum sistem ini menjelaskan tentang gambaran dari system yang akan dibuat. Pada gambaran sistem ini dibuat di sepeda motor Honda vario 125, kemudian kamera diletakkan didekat plat nomor untuk dapat mendeteksi keadaan jalan yang akan dilewati, hasil dari kamera tersebut kemudian dikirim ke mikrokomputer yang terletak di dalam *headlamp* sepeda motor itu sendiri. Mikrokomputer akan mengelolah data dari hasil kamera dan kemudian mengontrol lampu *led* yang disusun secara paralel yang terletak didalam *headlamp* tersebut untuk menyinari jalanan yang tidak terdapat pengendara lain.

4.1.1 Tujuan

Tujuan dari sistem yang akan dibangun ini adalah menjelaskan penelitian tentang membangun sistem pencahayaan pada sebuah sepeda motor untuk mengatur pencahayaan agar tidak mengganggu pengendara lain dan membuat pengguna kendaraan dapat melihat jalan lebih jelas dari sebelumnya.

Adapun penjelasan secara terperinci terhadap penelitian ini adalah fitur dari sistem yang dikembangkan, kegunaan, lingkungan pengguna, batasan perencanaan dan implementasi dan asumsi ketergantungan, serta penjelasan terkait kebutuhan fungsional dan non fungsional. Bab ini dibuat sebagai dokumentasi untuk para pengembang berikutnya untuk selanjutnya dapat diusulkan sebagai laporan skripsi.

4.1.2 Kegunaan

Dalam perancangan *headlamp* ini, nantinya pada *headlamp* tersebut dilengkapi dengan mikrokomputer, rangkaian stepdown, dan module kamera. Kegunaan rangkaian *stepdown* adalah untuk mengatur masukkan daya dari sepeda motor itu sendiri ke dalam sistem yang akan digunakan, karena pada daya yang diberikan oleh sepeda motor tersebut memiliki daya naik turun, sehingga penggunaan rangkaian *stepdown* ini juga berguna untuk menstabilkan daya tersebut agar rangkaian yang akan digunakan tidak rusak pada saat dijalankan. Kegunaan pada module kamera, berfungsi untuk menangkap aktifitas jalan yang dilalui, sehingga sistem dapat mengetahui dan mengontrol lampu sesuai dengan sistem yang diinginkan.

4.1.3 Karakteristik Pengguna

Karakteristik Pengguna pada sistem ini adalah sebagai pengguna kendaraan yang memiliki sebuah pencahayaan yang dapat mengatur pencahayaan secara bebas tanpa perlu mengganggu pengguna jalan yang lain, Pengguna dapat mematikan dan menghidupkan sistem ketika dibutuhkan.

4.1.4 Lingkungan Operasi

Lingkungan operasi merupakan persyaratan untuk mendukung berjalannya sistem ini yang meliputi waktu sistem ini dapat berjalan dengan baik ketika malam hari dengan kondisi tidak hujan dan peletakan kamera menghadap kedepan agar mampu mengontrol pencahayaan pada sistem yang dibangun.

4.2 Kebutuhan Antarmuka Eksternal

4.2.1 Kebutuhan Antarmuka Pengguna

Kebutuhan antarmuka pengguna dalam sistem ini adalah :

- a. Sistem dapat mengatur pencahayaan pada sepeda motor
Sistem secara otomatis mengatur pancaran cahaya lampu sepeda motor sehingga pengguna jalan lain yang berlawanan arah tidak silau.
- b. Sistem memiliki pencahayaan yang lebih terang dari sebelumnya
Pengguna dapat melihat jalan lebih jelas dari sebelumnya, karena dalam sistem ini memiliki pencahayaan yang lebih terang dengan menggunakan daya listrik yang lebih kecil dari pada pencahayaan aslinya.

4.2.2 Kebutuhan Perangkat Keras

Kebutuhan antarmuka perangkat keras yang terdapat pada sistem ini yaitu:

- a. Pada rangkaian *StepdownXL4015*
 - Membutuhkan daya perbandingan input dan output sebesar 0,5 V untuk dapat melakukan penurunan daya.
 - Memiliki arus maksimal sebesar 5A.
- b. Pada module kamera *Logitech C170*
 - Membutuhkan daya sebesar 5 V.
 - Komponen komunikasi menggunakan USB untuk mengirim hasil kamera ke mikrokomputer.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional adalah sebuah kebutuhan yang harus dipenuhi oleh sistem, berikut beberapa fungsional dari sistem ini yang harus bekerja dengan semestinya.

4.3.1 Dapat Membedakan Pengendara Lain Dengan Jalan

- a. Penjelasan
Fitur ini menjelaskan bagaimana sistem dapat mendeteksi pengendara lain dengan jalan. Untuk melakukan proses tersebut, pengguna hanya perlu mengaktifkan sistem dengan cara menekan tombol on pada tombol

yang telah disediakan. Apabila sudah di tekan, maka pengguna tidak perlu mengontrol lampu jauh lagi.

b. Stimulus atau Respon Sistem

Ketika tombol On aktif, maka sistem secara otomatis akan mendeteksi pengendara lain dengan menggunakan webcam dan akan diproses oleh raspberry pi dengan menggunakan opencv.

c. Kebutuhan fungsional

Fungsi ini digunakan sebagai cara agar pengguna tidak perlu lagi untuk mengontrol lampu jauh ketika sistem sedang di aktifkan.

4.3.2 Dapat Menerangi Jalan Lebih Baik Dari Sebelumnya

a. Penjelasan

Fungsi ini menjelaskan sistem menerangi bagian jalan yang gelap tanpa mengganggu pengendara lain, sehingga pengguna tidak perlu mengontrol lampu jauh ketika ada pengendara lain berada didepan pengguna maupun berlawanan arah ketika sistem aktif.

b. Stimulus atau Respon Sistem

Ketika sistem aktif, sistem secara otomatis mengaktifkan lampu ketika area yang dideteksi oleh *webcam* dirasa kurang cukup cahaya, sehingga pengguna dapat melihat area yang gelap tersebut.

c. Kebutuhan fungsional

Fungsi ini digunakan sebagai agar pengguna dapat melihat area jalan yang gelap tanpa perlu mengontrol lampu jauh.

4.4 Kebutuhan Non Fungsional

4.4.1 Kebutuhan Performansi

Sistem dapat bekerja dengan lebih cepat dengan resolusi video 320x240, karena semakin kecil resolusi video maka proses pengolahan citra digital menjadi semakin cepat.

4.4.2 Kebutuhan Keselamatan

Sistem dapat bekerja secara optimal dengan menggunakan tegangan 5 *Volt* dengan rentang arus 1,2 hingga 2,5 *Ampere* pada mikrokomputer. Tegangan dan arus yang tidak sesuai dengan semestinya akan menyebabkan kerusakan pada komponen.

4.5 Spesifikasi Perangkat Keras

Dalam membangun sistem pada penelitian ini diperlukan beberapa komponen perangkat keras. Berikut perangkat keras yang digunakan:

4.5.1 Laptop

Laptop yang digunakan untuk melakukan pemrograman pada mikrokomputer adalah laptop Toshiba Satellite L745, dapat dilihat pada gambar 4.1



Gambar 4.1Laptop Thosiba L745

Berikut ini adalah spesifikasi Laptop Thosiba L745 yang ditunjukkan pada Tabel 4.1.

Tabel 4.1Spesifikasi Laptop Thosiba L745

Laptop Thosiba L745	
Tipe	Netbook
Nama Produk	Thosiba L745
Memory	4 GB
Storage	300 GB
Graphic	Intel HD
Sistem Operasi	Windows 10

4.5.2 Mikrokomputer

Mikrokomputer yang digunakan pada sistem ini adalah *RaspberryPi 3 Model B*. Tampilan dari mikrokomputer *RaspberryPi 3 model B* dapat dilihat pada gambar 4.2



Gambar 4.2*Raspberry Pi 3 Model B*

Berikut ini spesifikasi Mikrokomputer *RaspberryPi 3* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Spesifikasi Mikrokomputer *Raspberry Pi 3*

Mikrokomputer <i>Raspberry Pi 3</i>	
Nama Produk	<i>Raspberry Pi 3, Model B</i>
Chip	<i>Broadcom BCM2837</i>
Core Architecture	<i>Quad-core ARM Cortex-A53</i>
Central Processing Unit	<i>1.2 GigaHertz</i>
Graphic Processing Unit	<i>Broadcom VideoCore IV</i>
Memory	<i>1 GB LPDDR2</i>
Sistem Operasi	<i>Raspbian</i>
Dimensi	<i>85 x 56 x 17 mm</i>
Catu Daya	<i>Mikro USB socket 5V, 2A</i>

4.5.3 WebcamLogitech C170

Karena *Webcam* yang digunakan dalam menangkap gambar jalan yang akan dilalui adalah kamera *WebcamLogitech C170*. Berikut ini tampilan *Logitech C170* yang ditunjukkan pada gambar 4.3



Gambar 4.3 Logitech C170

Kamera *webcamLogitech C170* memiliki spesifikasi yang dapat dilihat pada tabel 4.3.

Tabel 4.3 Spesifikasi *Webcam Logitech C170*

<i>Webcam Logitech C170</i>	
Resolusi Video	640 x 480 pixels
Resolusi Gambar	1024 x 768 pixels
Interface to PC	<i>High Speed USB 2.0</i>
Operasting System Support	Windows (XP / Vista / 7) & Linux

4.5.4 *LedCreeX-TE*

Led yang digunakan dalam penerangan jalan adalah *LedCreeX-TE 5W*. Berikut tampilan *ledcree X-TE 5W* dapat dilihat pada gambar 4.4



Gambar 4.4*Led Cree X-TE 5W*

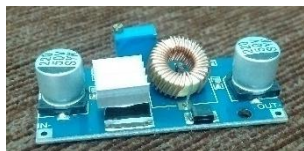
Led Cree X-TE 5W memiliki spesifikasi yang dapat dilihat pada tabel 4.4

Tabel 4.4*Spesifikasi Led Cree X-TE 5W*

<i>Led Cree X-TE 5W</i>	
Sumber	3,3 – 3,5 V
Arus	1,5 A
<i>Lumens</i>	107 – 456
Dimensi	2 x 3,45 x 3,45 mm
<i>View Angle</i>	115 derajat
Diameter	16 mm

4.5.5 Stepdown XL4015

Stepdown yang digunakan untuk menurunkan tegangan dari sepeda motor adalah stepdown XL4015 DC – DC. Berikut tampilan *stepdown* XL4015 DC – DC dapat dilihat pada Gambar 4.5



Gambar 4.5*Stepdown XL4015 DC – DC*

Stepdown XL4015 DC – DC memiliki spesifikasi yang dapat dilihat pada tabel 4.5.

Tabel 4.5*Spesifikasi Stepdown XL4015 DC – DC*

<i>Stepdown XL4015 DC – DC</i>	
Tegangan Input	8 – 36 V
Tegangan Output	1,25 – 36 V
Arus Maksimal	5 A

4.5.6 Transistor

Transistor yang digunakan untuk menyambung dan memutuskan arus adalah *transistor* BD-139, gambar dapat dilihat pada Gambar 4.6



Gambar 4.6*Transistor BD-139*

Berikut ini spesifikasi dari *transistor* BD-139 ditunjukkan pada tabel 4.6

Tabel 4.6Spesifikasi *Transistor BD-139*

<i>Transistor BD-139</i>	
<i>Supplier Package</i>	<i>BD-139</i>
<i>Category</i>	<i>Power MOSFET</i>
<i>Maximum drain source voltage</i>	<i>55V</i>
<i>Maximum continuous drain current</i>	<i>110A</i>
<i>Maximum power dissipation</i>	<i>200000mW</i>
<i>Maximum gate source voltage</i>	<i>±20V</i>
<i>Maximum drain source resistance</i>	<i>0,008Ohm@10V</i>

BAB 5 PERANCANGAN DAN IMPLEMENTASI

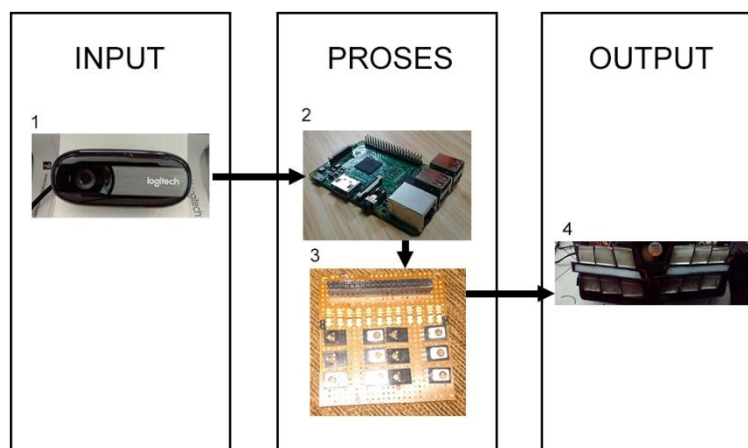
Pada tahapan perancangan sistem ini dijelaskan tentang konsep dasar, gambaran umum, analisis kebutuhan, diagram blok, perancangan perangkat keras, dan perancangan perangkat lunak sistem.

5.1 Perancangan Sistem

Pada tahap perancangan sistem ini menjelaskan tentang konsep dasar mengenai gambaran umum sistem yang akan dibuat, dan membahas perancangan pada aspek perangkat keras dan perangkat lunak.

5.1.1 Gambaran Umum Sistem

Perancangan sistem dan implementasi *headlamp* pada sepeda motor sebagai pengatur jarak aman pancaran cahaya dengan menggunakan sistem *ledmatrix* terdiri dari tiga komponen utama perangkat keras, yaitu kamera yang menghasilkan video sebagai *input*, mikrokomputer **Raspberry Pi 3** sebagai pengolah data, serta kumpulan *led* sebagai *output* dari sistem. Gambaran umum sistem dapat dilihat pada gambar 5.1.



Gambar 5.1 Gambaran Umum Sistem

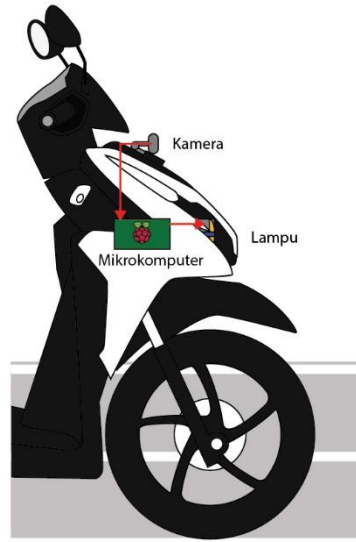
Keterangan

1. Camera Webcam
2. Raspberry Pi
3. Rangkaian Transistor
4. Led

Pada gambar 5.1 menunjukkan tentang gambaran umum sistem dimana sebuah kamera digunakan untuk mendapatkan kondisi jalan yang akan diproses oleh *RaspberryPi*. Setelah data kamera diproses oleh *RaspberryPi*, kemudian *RaspberryPi* mengontrol *led* dengan menggunakan rangkaian *transistor*.

5.1.2 Perancangan Perangkat Keras

Dalam perancangan perangkat keras yang meliputi rangkaian mikrokomputer yang didalamnya terdapat modul kamera. Setelah itu terdapat rangkaian *transistor* yang meliputi *led* untuk digunakan dalam pengontrolan lampu. Untuk lebih jelasnya seperti gambar dibawah ini.

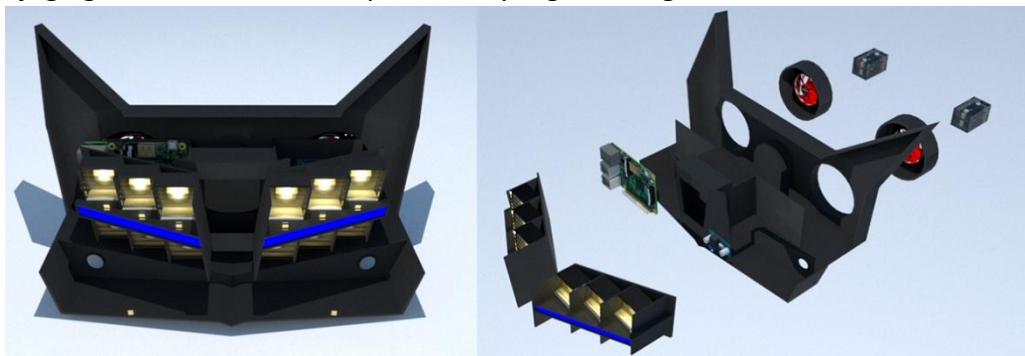


Gambar 5.2Desain pada sepeda motor

Pada Gambar 5.2 menjelaskan tentang perancangan perangkat keras yang akan dikerjakan, dimana kamera diletakkan didepan, kemudian untuk *mikrokomputer* dan lampu yang digunakan berada didalam *headlamp* itu sendiri.

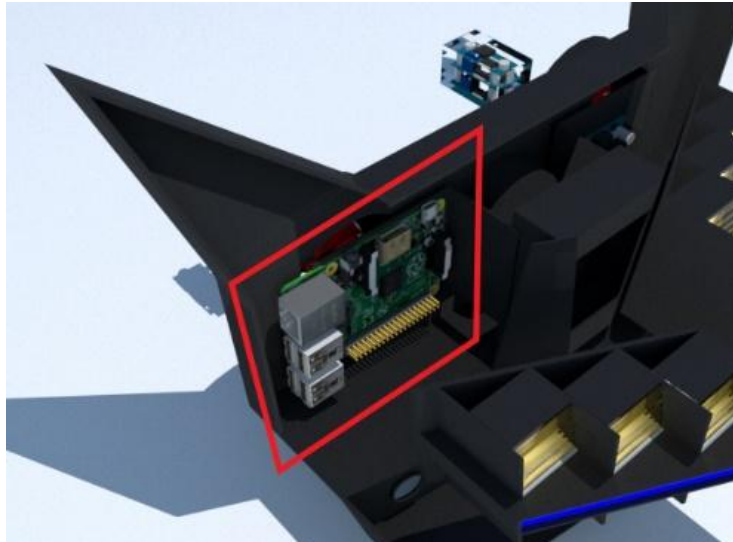
5.1.2.1 Perancangan Frame Lampu

Perancangan desain mekanik pada lampu utama yang akan digunakan menggunakan aplikasi *SketchUp*, di desain ini tercantum semua tempat bagian seperti: mikrokomputer, rangkaian *stepdown*, kipas pendingin, penempatan *led*, dan juga gambar akhir dari lampu utama yang akan digunakan.



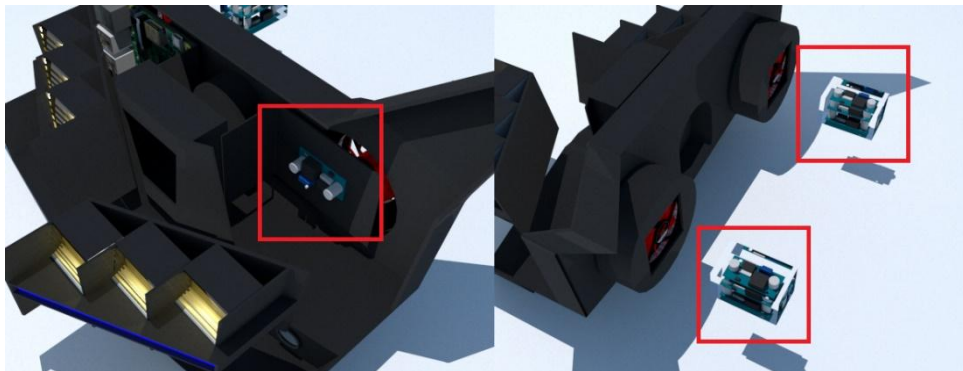
Gambar 5.3Desain Akhir Lampu Utama

Pada gambar 5.3 menggambarkan keseluruhan bagian lampu utama yang akan digunakan. Pada gambar tersebut merupakan gambar hasil ketika digunakan pada sepeda motor yang digunakan.



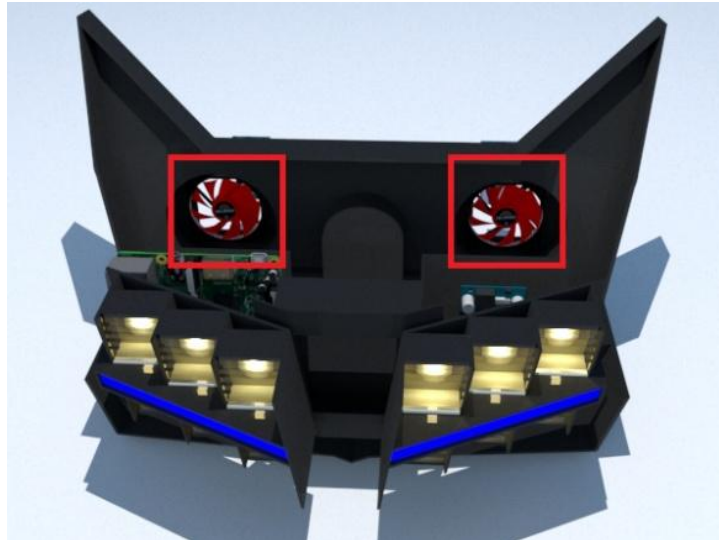
Gambar 5.4Desain Peletakan Mikrokomputer

Pada gambar 5.4 menggambarkan tentang peletakan mikrokomputer pada *headlamp* yang akan digunakan. Penempatan mikrokomputer tersebut bertujuan untuk memaksimalkan ruang kosong yang berada di dalam *headlamp*.



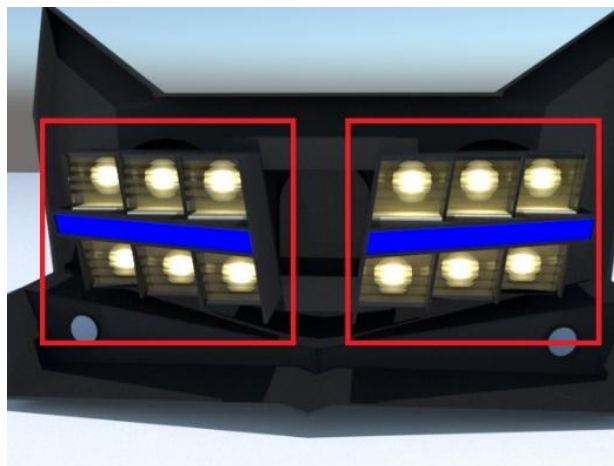
Gambar 5.5Desain Peletakan Rangkaian *Stepdown*

Pada gambar 5.5 menggambarkan tentang peletakan rangkaian *stepdown*, peletakan rangkaian *stepdown* ada dua lokasi, yaitu satu di dalam *headlamp*, dan yang lainnya berada diluar *headlamp*. Untuk rangkaian *stepdown* yang berada didalam berguna untuk memberikan daya ke mikrokomputer, sedangkan rangkaian *stepdown* lainnya berguna untuk memberi daya langsung ke sejumlah *led*.



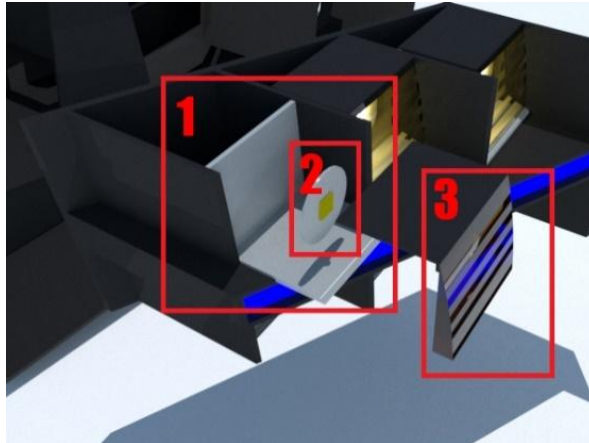
Gambar 5.6Desain Peletakan Kipas Pendingin

Pada gambar 5.6 menggambarkan tentang peletakan kipas pendingin yang digunakan, kegunaan dari kipas pendingin sendiri untuk mendinginkan bagian dalam *headlamp* itu sendiri.



Gambar 5.7Desain Pelatakan *Led Matrix*

Pada gambar 5.7 menggambarkan tentang peletakan lampu yang akan digunakan. Pada bagian lampu tersebut memiliki tiga bagian, dapat dijelaskan pada gambar 5.8.



Gambar 5.8Desain Bagian Lampu

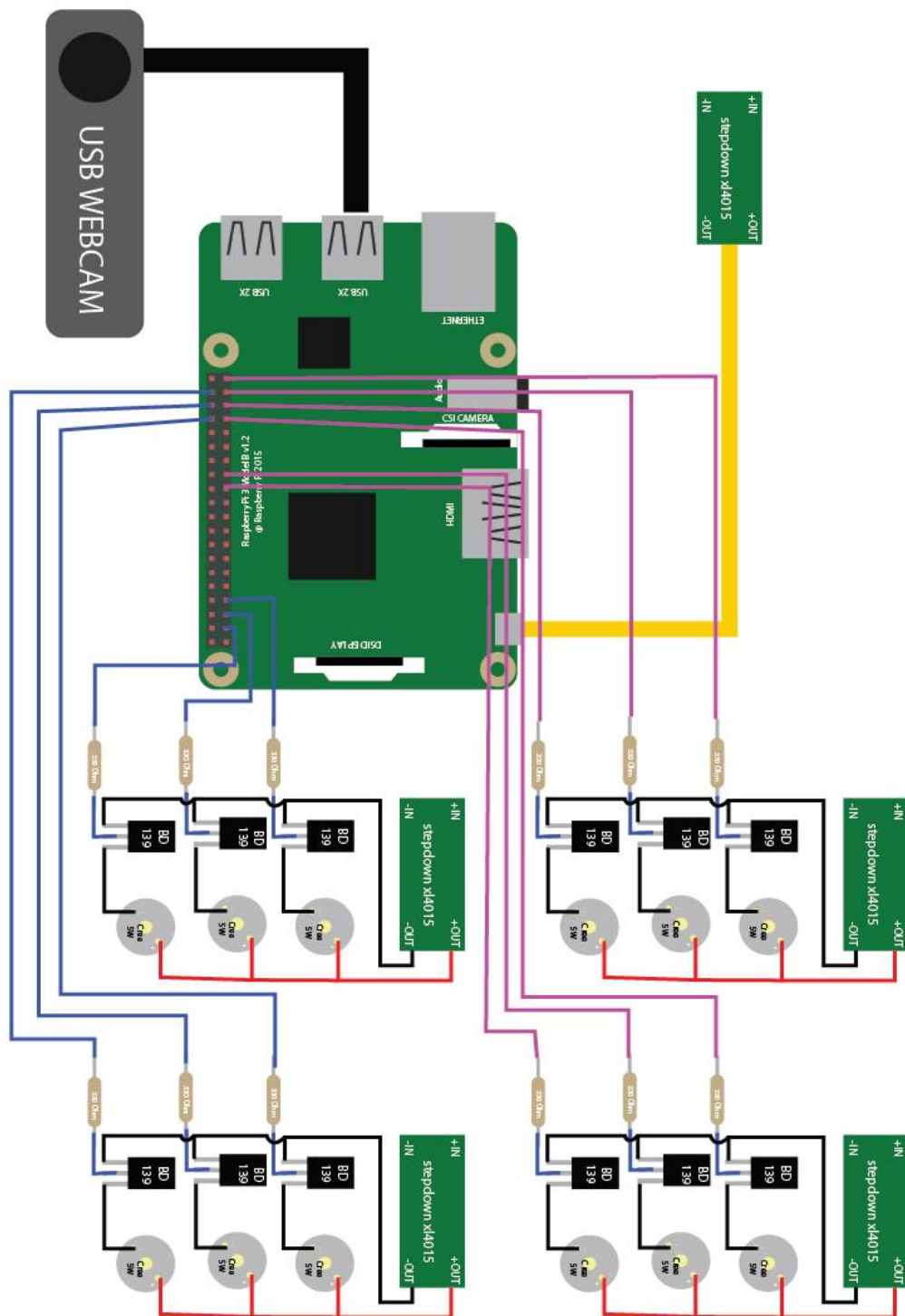
Keterangan:

1. *Heatsink*
2. *Led*
3. Lensa

Pada desain lampu tersebut memiliki fungsi yang berbeda-beda. *Heatsink* digunakan untuk mendinginkan lampu *led* yang memiliki suhu cukup panas ketika sedang beroperasi. *Led* digunakan untuk menerangi jalan yang akan dilalui. Dan untuk lensa sendiri digunakan untuk membuat cahaya yang dihasilkan oleh *led* dapat menjangkau lebih jauh dari sebelumnya.

5.1.2.2 Perancangan Rangkaian Alat

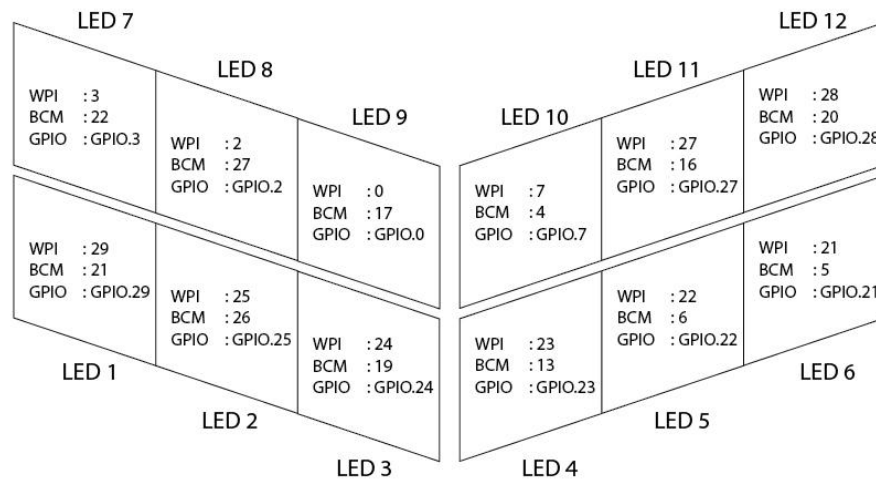
Pada perancangan rangkaian alat ini dijelaskan pada gambar 5.9.



Gambar 5.9 Skematik Rangkaian Alat

Pada gambar 5.9 dijelaskan bahwa untuk sistem dalam sistem lampu tersebut digunakan semua rangkaian sederhana, yaitu penggunaan *stepdown* sebagai pengatur daya keseleuran dengan menggunakan 5 *stepdown* yang diatur dengan daya berbeda-beda, dimana untuk *RaspberryPi* sendiri digunakan dengan daya 5v sedangkan untuk *led* menggunakan daya 3,3v, lalu menggunakan

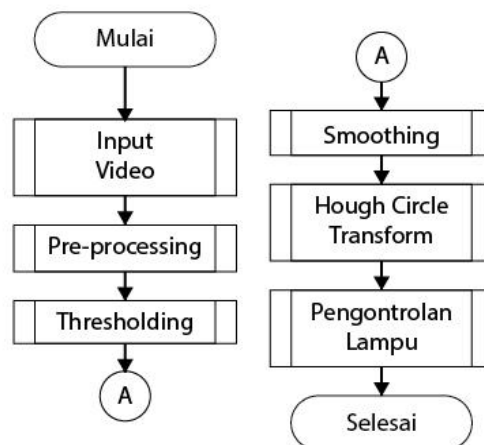
transistor pada tiap *led* yang akan digunakan untuk mematikan dan menhidupkan *led*. Pada *webcam* digunakan sebuah *usb* untuk koneksinya. Untuk susunan *led* yang digunakan dapat dilihat pada gambar 5.10.



Gambar 5.10Susunan Led pada Headlamp

5.1.3 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak membahas tentang diagram alir sistem dalam mendeteksi kendaraan lain dan mengontrol lampu serta proses yang dilakukan pada masing-masing alir tersebut. Berikut ini rancangan diagram alir yang akan ditunjukkan pada gambar 5.11.

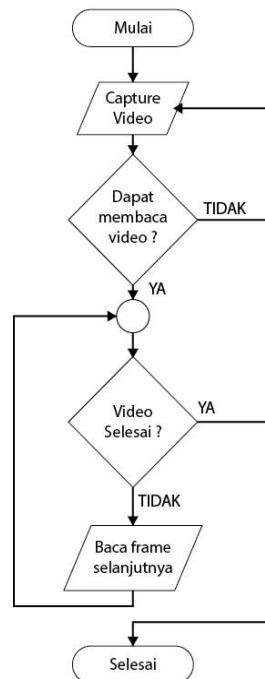


Gambar 5.11Diagram Alir Sistem

Pada gambar 5.11 terdapat beberapa proses yang harus dilakukan dalam sistem yang terdiri dari *input video*, *pre-processing*, *Thresholding*, *Smoothing*, *HoughCircleTransform*, dan Pengontrolan lampu.

5.1.3.1 Input Video

Input video adalah merupakan tahap pertama saat sistem memulai menjalankan program. *Input* video digunakan untuk memasukkan data video sebagai data yang akan dikelola. Pada tahap ini data masih berupa video akan di kelolah menjadi beberapa frame dan disimpan ke dalam bentuk matrik. Berikut *flowchart* proses input video ditunjukkan pada gambar 5.12.

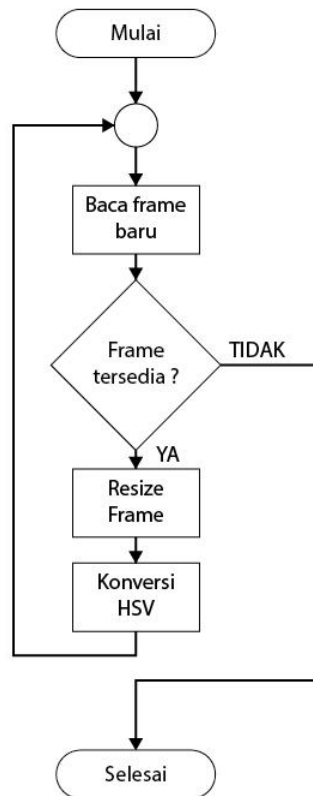


Gambar 5.12*Flowchart* *Input Video*

Pada gambar 5.12 menjelaskan langkah pertama yang dilakukan adalah *capture* video kemudian memeriksa sistem apakah dapat membaca video atau tidak. Jika sistem tidak dapat membaca video maka program akan mengambil video lagi. Selanjutnya adalah program akan membaca setiap frame terbaru hingga video yang di *capture* selesai.

5.1.3.2 Pre-processing

Pada tahap Pre-processing digunakan untuk mempersiapkan citra yang akan dikelola untuk proses selanjutnya. Kondisi citra sebelum proses ini masih berukuran sama dengan resolusi video. Terdapat beberapa proses yang akan dilakukan pada tahan *pre-processing* yaitu *resizeframe* dan konversi HSV. Berikut ini *flowchart* tahap *pre-processing* yang ditunjukkan pada gambar 5.13.



Gambar 5.13FlowchartPre-processing

Pada gambar 5.13 setelah sistem berhasil meng *input* video maka program membaca *frame* secara berurutan. Apabila *frame* sudah tidak tersedia maka program selesai. *Frame* yang dapat dibaca kemudian dilakukan *resize* sehingga ukuran tiap *frame* menjadi 320x240 dengan menggunakan *interpolasiliner*. *Interpolasilinear* adalah sebuah metode yang digunakan untuk mengetahui nilai dari suatu titik dalam garis lurus. Proses *resizeframe* digunakan untuk dapat memperingan dalam pengolahan data karena semakin kecil ukuran *frame* maka kecepatan pengelolah data semakin cepat.

5.1.3.3 Thresholding

Proses Thresholding digunakan untuk menghasilkan citra biner dengan melakukan konversi terhadap citra HSV. Proses ini dilakukan dengancara merubah nilai piksel dengan nilai yang telah ditentukan agar objek yang diinginkan dapat dibedakan dengan objek lainnya.Tabel 5.1 adalah jangkauan nilai objek yang yang digunakan.

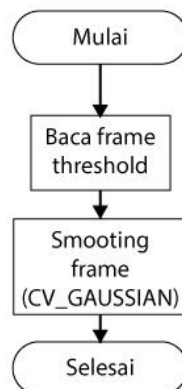
Tabel 5.1 Kesimpulan jangkauan warna objek yang diinginkan

	H	S	V
Min	0	0	200
Max	180	255	255

Pada Tabel 5.1 dianalisis dengan menerapkan jangkauan warna pada HVS yakni, H(0-239),S(0-240),V(0-240). Dengan begitu dapat digunakan untuk menentukan pendeteksian objek yang diinginkan dan memasukkan ke dalam program yang digunakan.

5.1.3.4 Smoothing

Proses Smoothing digunakan untuk menghilangkan *noise* akibat konversi dari RGB ke HSV, dengan tujuan mengurangi error saat pendeteksi warna. Berikut ini *flowchart* tahap smoothing yang ditunjukkan pada gambar 5.14.

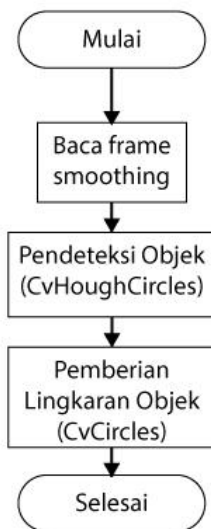


Gambar 5.14 Flowchart Smoothing

Fungsi ini adalah implementasi dari sistem smoothing, dimana *frame* sebelumnya di ambil kemudian *frame* tersebut diproses dengan sistem CV_GAUSSIAN. Proses *GaussianBlur* menggunakan proses konvolusi antara nilai piksel pada frame asli dengan nilai *Gaussian* pada *frame* kernel. Proses ini dijelaskan pada sub bab 2.2.5.

5.1.3.5 HoughCircleTransform

Fungsi *HoughCircle* sudah dijelaskan pada sub bab 2.2.6 tentang metode pendeteksian *HoughCircle* secara umum. Pada OpenCV fungsi ini sedikit berbeda karena hanya menggunakan gradasi warna sebagai ukurannya.



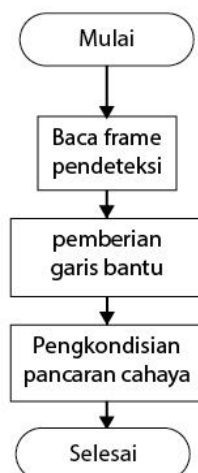
Gambar 5.15*FlowchartHoughCircleTransform*

Dari gambar 5.15 menjelaskan proses pada *houghcircletransform*, dimana pada pendeteksian objek disediakan parameter-parameter tambahan seperti resolusi pendeteksian, minimal jarak antar titik centroid, batas atas dan bawah thresholding, serta parameter optional yakni maksimum dan minimum jari-jari, karena tidak berpengaruh terhadap jumlah lingkaran dari perubahan koordinat centroid.

Kemudian pada bagian pemberian lingkaran objek digunakan dengan menggunakan hasil ukuran pendeteksi objek sebagai lokasi objek yang kemudian diberi lingkaran dengan menggunakan *cvcircles*.

5.1.3.6 Pengontrolan Lampu

Fungsi pengontrolan lampu digunakan untuk mengatur agar pengendara lain tidak silau. Berikut adalah flowchart untuk membuat kotak.



Gambar 5.16*PreudeoCode Pembuatan Garis*

Dari gambar 5.16, sistem pengontrolan lampu perlu adanya garis bantu dalam pengalokasi pancaran cahaya yang akan dibuat, kemudian dari pembuatan garis bantu menjadikan beberapa bentuk kotak yang digunakan sebagai perwakilan pancaran cahaya tiap led. Kemudian setelah mengetahui lokasi pancaran cahaya tersebut, maka tinggal memberikan kondisi tiap *led* yang telah tersusun dengan mengikuti kotak yang telah dibuat sebelumnya.

5.2 Implementasi Sistem

Pada tahapan implementasi sistem menjelaskan tentang batasan-batasan implementasi, implementasi perangkat keras, dan implementasi perangkat lunak.

5.2.1 Implementasi Perangkat Keras

Pada implementasi perangkat keras memiliki beberapa bagian yang akan dijelaskan dibawah ini:

5.2.1.1 Implementasi sistem lampu sepeda motor

Implementasi sistem lampu sepeda motor yang digunakan adalah *RaspberryPi* 3 dan yang digunakan hanyalah sebuah kamera *webcam* yang memberikan data berupa video. Dari hasil video tersebut akan dikelola oleh *RaspberryPi* 3 untuk menentukan objek kendaraan dengan cara memecah video menjadi beberapa *frame* yang kemudian diproses untuk menentukan objek. Setelah itu dari hasil proses tersebut, *RaspberryPi* 3 akan mengontrol sejumlah *ledcree* yang disusun secara paralel dengan menggunakan *transistor* BD139 untuk mematikan dan menghidupkan lampu agar pengendara lain tidak silau. Untuk lebih jelasnya dapat dilihat pada gambar 5.17.



Gambar 5.17 Implementasi Lampu Sepeda Motor

5.2.2.2 Implementasi peletakan kamera

Implementasi peletakan kamera pada sepeda motor yang digunakan adalah kamera *webcam* Logitech c170 yang akan diletakkan didepan sepeda motor. Peletakan tersebut bertujuan untuk memonitori keadaan jalan yang akan dilalui. Untuk lebih jelasnya dapat dilihat seperti gambar 5.18.



Gambar 5.18 Implementasi Peletakan Kamera

5.2.2.3 Implementasi desain *headlamp*

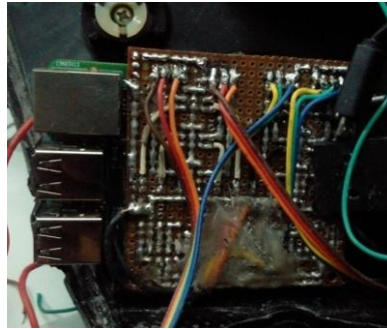
Implementasi desain headlamp yang telah didesain sebelumnya bertujuan untuk memaksimalkan ruang yang ada di *headlamp* sepeda motor tersebut agar tidak merubah kondisi sepeda motor tersebut. Berikut hasil implementasi dari desain 3 Dimensi menjadi hasil sebenarnya pada gambar 5.19.



Gambar 5.19 Implementasi Desain Headlamp

5.2.2.4 Implementasi Mikrokomputer dan Rangkaian *Transistor*

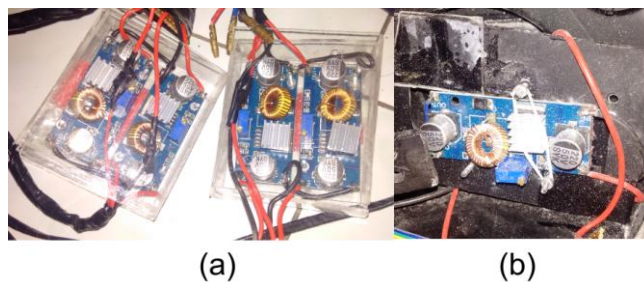
Implementasi mikrokomputer dan rangkaian *transistor* yang berada didalam *headlamp* sepeda motor tersebut. Penggabungan 2 rangkaian ini bertujuan agar tidak adanya kesalahan dalam pemasangan rangkaian transistor ketika sedang instalasi pada *headlamp* sepeda motor. Implementasi dapat dilihat pada gambar 5.20.



Gambar 5.20 Implementasi Mikrokomputer dan Rangkaian *Transistor*

5.2.2.5 Implementasi *Stepdown* XL4015

Pada implementasi pengatur daya, digunakan 5 buah *stepdown* XL4015 yang bertujuan untuk memberi daya yang sesuai dengan kebutuhan alat tersebut. Sebuah *stepdown* XL4015 di atur dengan daya sebesar 5v yang digunakan untuk *RaspberryPi* dan 4 buah lainnya digunakan untuk daya *led*. Implementasi dapat dilihat pada gambar 5.21.



Gambar 5.21 Implementasi *Stepdown* (a) untuk *RaspberryPi* (b) untuk *Led*

5.2.2.6 Implementasi *Led*

Pada implementasi *led* dengan jumlah 12 buah yang disusun secara paralel. Dengan tatahan 3 buah *led* untuk sebuah *stepdown*. Peletakan led disusun agar dapat mewakili tiap pancaran yang diinginkan sehingga dapat memudahkan penulis dalam mengalokasikan pancaran cahaya tersebut. Berikut hasil implementasi *led* yang dapat dilihat pada gambar 5.22.



Gambar 5.22 Implementasi *Led*

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat keras dilakukan berdasarkan perancangan sistem. Pada bagian ini membahas implementasi *input video*, *pre-processing*, *Thresholding*, *Smoothing*, *Hough Circle Transform*, dan Pengontrolan lampu.

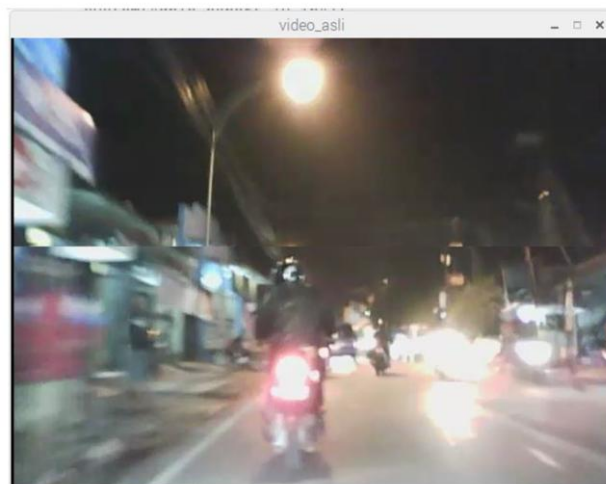
5.2.2.1 Input Video

Implementasi input video menghasilkan data video yang dapat diproses pada tahap selanjutnya. Berikut ini pseudo code input video yang ditampilkan pada tabel 5.2.

Tabel 5.2Potongan Program Input Video

46	<code>CvCapture* video;</code>
47	<code>Video = cvCaptureFromCAM(0);</code>
48	<code>cvGrabFrame(video);</code>
49	<code>IplImage * gambar_asli = cvRetrieveFrame(video);</code>
50	
51	<code>If (!gambar_asli) {</code>
52	<code> Cout << "Error: cap webcam not accessed successfully\n\n";</code>
53	<code> Return(0);</code>
54	<code>}</code>

Tabel 5.2 menunjukkan pengambilan video dengan menggunakan kamera, ketika video telah diperoleh lalu diambil perframe dengan menggunakan *cvRetrieveFrame*. Kemudian cek *frame* apakah berhasil di ambil atau tidak dengan memberi kondisi jika tidak terdeketsi akan menampilkan pesan error. Berikut Hasil dari potongan program terdapat pada gambar 5.23.



Gambar 5.23Hasil Input Video

5.2.2.2 Pre-processing

Implementasi pre-processing menghasilkan tampilan *input* video yang telah diatur ukurannya dan telah dikonversi citra. Berikut ini *pseudocodepre-processing* yang ditampilkan pada tabel 5.3.

Tabel 5.3 Potongan Program *Pre-processing*

71	<code>IplImage *ukuran_gambar = cvCreateImage(cvSize(320, 240),8,3);</code>
72	<code>cvResize(gambar_asli,ukuran_gambar,CV_INTER_LINEAR);</code>
73	<code>IplImage *HSV_putih =</code> <code>cvCreateImage(cvGetSize(ukuran_gambar),IPL_DEPTH_8U, 3);</code>
74	<code>cvCvtColor(ukuran_gambar, HSV_putih, CV_BGR2HSV);</code>

Tabel 5.3 Menunjukkan proses dalam mengubah ukuran *frame* yang diatur dengan ukuran 320 x 240. Kemudian *frame* yang telah berukuran lebih kecil dari sebelumnya lalu di konversikan dari citra RGB menjadi citra HSV dengan menggunakan *sintaxcvCvtColor*. Berikut Hasil dari potongan program terdapat pada gambar 5.24 dan gambar 5.25.



Gambar 5.24 Hasil *Resize* ,(a) Sebelum *Resize*, (b) Sesudah *Resize*



Gambar 5.25 Hasil *Conversi Citra*, (a) sebelum *diconversi*, (b) sesudah *dikonversi*

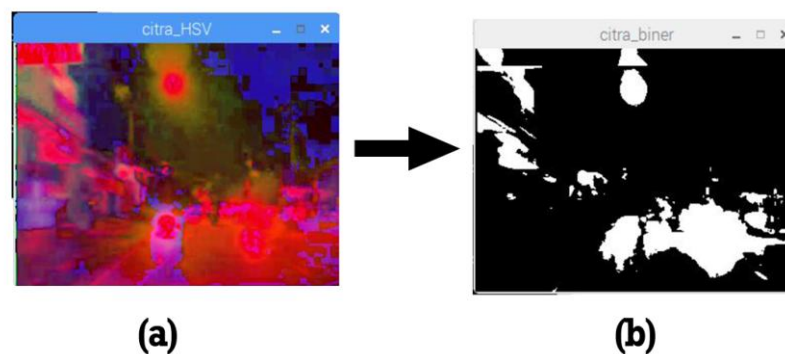
5.2.2.3 Thresholding

Implementasi thresholding menghasilkan citra HSV dalam bentuk citra biner. Berikut ini *pseudocode* thresholding citra frame yang ditampilkan pada tabel 5.4.

Tabel 5.4 Potongan Program *Thresholding*

21	<code>IplImage *putih10 = cvCreateImage(cvGetSize(HSV_putih),IPL_DEPTH_8U, 1);</code>
22	<code>cvInRangeS(HSV_putih,cvScalar(0,0,200,0),cvScalar(180,255,255,0),p utih10);</code>

Tabel 5.4 Menunjukkan proses *thresholding* dengan menentukan minimal dan maksimal dalam pencarian warna *thresholding* yang diinginkan. Berikut Hasil dari potongan program terdapat pada gambar 5.26.



Gambar 5.26 Hasil Thresholding

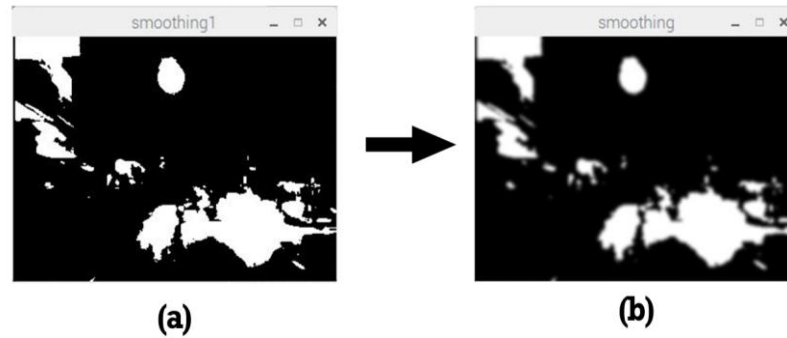
5.2.2.4 Smoothing

Implementasi *Smoothing* menghasilkan objek yang sebelumnya berwarna padat menjadi buram untuk menghilangkan objek yang tidak diinginkan. Berikut ini *pseudocode* smoothing yang ditunjukkan pada tabel 5.5.

Tabel 5.5 Potongan Program *Smoothing*

79	<code>cvSmooth(putih10,putih10,CV_GAUSSIAN,9,9);</code>
----	---------------------------------------------------------

Tabel 5.5 Menunjukkan *pseudocode* dengan memanggil fungsi CV_GAUSSIAN dalam melakukan *smoothing*. Berikut Hasil dari potongan program terdapat pada gambar 5.27.



Gambar 5.27Hasil Smoothing

5.2.2.5 *HoughCircleTransform*

Implementasi *HoughCircleTransform* digunakan untuk mendeteksi adanya objek dideteksi, pendeteksian dijelaskan pada tabel 5.6.

Tabel 5.6Potongan Program *HoughCircleTranform*

81	<code>p_seqCirclesputih10 = cvHoughCircles(</code>
82	<code>putih10, //input frame</code>
83	<code>simpan_putih10, //memoru untuk elements</code>
84	<code>CV_HOUGH_GRADIENT, //algoritma berbasis gradien warna</code>
85	<code>2, //resolusi gradient pixel</code>
86	<code>40, //minimal jarak</code>
87	<code>255, //batas atas canny threshold</code>
88	<code>40, //batas bawah canny threshold</code>
89	<code>8, //min jari-jari lingkaran dalam pixel</code>
90	<code>40); //max jari-jari lingkaran dalam pixel</code>

Tabel 5.6, fungsi pendeteksian dinyatakan deklarasi parameter *HoughCircle*. Dalam parameter tersebut terlihat bahwa memori `simpan_putih10` digunakan sebagai memori penyimpanan element pendeteksian *HoughCircle*. Algoritma ini adalah satu-satunya algoritma yang disediakan oleh OpenCV.

Kemudian didefinisikan resolusi *gradient*, sebagai parameter untuk tingkat akurasi pendeteksian warna berbasis *gradient*. Kemudian didefinisikan juga jarak antara titik tengah lingkaran dengan titik tengah lingkaran lain sebagai aturan jarak antar objek lingkaran terdeteksi yang diperbolehkan, setelah itu didefinisikan nilai ambang atas *threshold* dan ambang bawah *threshold*, dengan metode *CannyThreshold*. *CannyThreshold* adalah metode untuk menentukan batas minimal dan maksimal suatu threshilding pada gambar. Kemudian didefinisikan ukuran radius minimal dan maksimal dalam pendeteksian lingkaran. Radius minimal dan maksimal pada lingkaran digunakan hanya sebagai opsi saja, karena tidak berpengaruh pada banyaknya objek terdeteksi.

Fungsi berikutnya adalah mendapat titik *centroid* dan jari-jari lingkaran pada *listing* tabel 5.7.

Tabel 5.7 Potongan Program Titik Centroid

93	for(i=0; i <p_seqCirclesputih10->total; i++){
94	
95	KorXYRputih10 = (float*)cvGetSeqElem(p_seqCirclesputih10, i);
96	
97	cvCircle(ukuran_gambar, // gambar di frame
98	cvPoint(cvRound(KorXYRputih10[0]),
99	cvRound(KorXYRputih10[1])), //tengah centroid
100	3, // 3 pixel jari lingkarn
101	CV_RGB(0,100,0), // warna hijau
102	CV_FILLED); //diisi dengan ketebalan 3 pixel
103	cvCircle(ukuran_gambar,cvPoint(cvRound(KorXYRputih10[0]),
104	cvRound(KorXYRputih10[1])),cvRound(KorXYRputih10[2]),CV_RGB(0,100,0),3);

Pada tabel 5.7 dijelaskan *listing* fungsi mendapatkan *centroid* dengan menggunakan iterasi (kolong *for*). Dalam fungsi ini dibicarakan sebagai *cvGetSeqElem*, untuk mendapatkan 3 parameter *HoughCircles*, yakni titik pusat centroid (x,y), dan jari-jari lingkaran. Setelah mendapat 3 parameter tersebut, dapat dibatasi pendeteksian dengan menggunakan garis bantu. Berikut *pseudocode* dalam membuat garis bantu. Berikut Hasil dari potongan program terdapat pada gambar 5.28.



Gambar 5.28 Hasil Hough Circle Tranform, (a) Frame Tanpa Hough Circle Tranform, (b) Frame Dengan Hough Circle Tranform

Tabel 5.8 Potongan Program Pembuatan Garis Bantu

105	//Buat garis bantu
106	//garis vertikal
107	cvLine(ukuran_gambar, Point(0, 140), Point(320, 140), Scalar(255, 255, 255), 1, 8);
108	cvLine(ukuran_gambar, Point(0, 210), Point(320, 210), Scalar(255, 255, 255), 1, 8);
109	
110	//garis horizontal
111	cvLine(ukuran_gambar, Point(53, 140), Point(53, 210), Scalar(255, 255, 255), 1, 8);

112	cvLine(ukuran_gambar, Point(106, 140), Point(106, 210), Scalar(255, 255, 255), 1, 8);
113	cvLine(ukuran_gambar, Point(159, 140), Point(159, 210), Scalar(255, 255, 255), 1, 8);
114	cvLine(ukuran_gambar, Point(212, 140), Point(212, 210), Scalar(255, 255, 255), 1, 8);
115	cvLine(ukuran_gambar, Point(265, 140), Point(265, 210), Scalar(255, 255, 255), 1, 8);

Pada tabel 5.8 menjelaskan fungsi membuat garis bantu dengan menentukan titik awal garis dan titik akhir garis, kemudian menentukan warna garis.

5.2.2.6 Pengontrolan Lampu

Implementasi pada pengontrolan lampu menghasilkan variasi lampu yang berbeda-beda agar tidak menyilaukan pengendara lain. Berikut *pseudocode* yang digunakan dalam mengondisikan *led*.

Tabel 5.9 Potongan Program pengondisian *Led*

120	digitalWrite (21, HIGH) ;
121	digitalWrite (26, HIGH) ;
122	digitalWrite (19, HIGH) ;
123	digitalWrite (13, HIGH) ;
124	digitalWrite (6, HIGH) ;
125	digitalWrite (5, HIGH) ;
126	if(KorXYRputih10[0]>=0 && KorXYRputih10[0]<=53 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){
127	digitalWrite (20, LOW) ;}
128	else if(KorXYRputih10[0]>53 && KorXYRputih10[0]<=106 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){
129	digitalWrite (16, LOW) ;}
130	else if(KorXYRputih10[0]>106 && KorXYRputih10[0]<=159 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){
131	digitalWrite (4, LOW) ;}
132	else if(KorXYRputih10[0]>159 && KorXYRputih10[0]<=212 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){
133	digitalWrite (22, LOW) ;}
134	else if(KorXYRputih10[0]>212 && KorXYRputih10[0]<=265 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){
135	digitalWrite (27, LOW) ;}
136	else if(KorXYRputih10[0]>=265 && KorXYRputih10[0]<=320 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){
137	digitalWrite (17, LOW) ;}
138	
139	else{
140	
141	digitalWrite (22, HIGH) ;

142	<code>digitalWrite (27, HIGH) ;</code>
143	<code>digitalWrite (17, HIGH) ;</code>
144	<code>digitalWrite (4, HIGH) ;</code>
145	<code>digitalWrite (16, HIGH) ;</code>
146	<code>digitalWrite (20, HIGH) ;}</code>

Pada tabel 5.9, menjelaskan pengondisian led dengan menggunakan area yang telah dibuat sebelumnya dengan garis bantu, pengondisian *led* akan mati ketika area *led* tersebut terdeteksi objek.

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk menganalisis apakah sistem telah bekerja sesuai perancangan. Pengujian dilakukan perblok kemudian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk mempermudah analisa apabila sistem ini tidak bekerja sesuai perancangan. Pengujian yang perlu dilakukan adalah sebagai berikut:

1. Pengujian pendeteksi kendaraan
2. Pengujian pengontrolan lampu
3. Pengujian validitasi sistem

6.1 Pengujian Pendeteksi Kendaraan

6.1.1 Tujuan

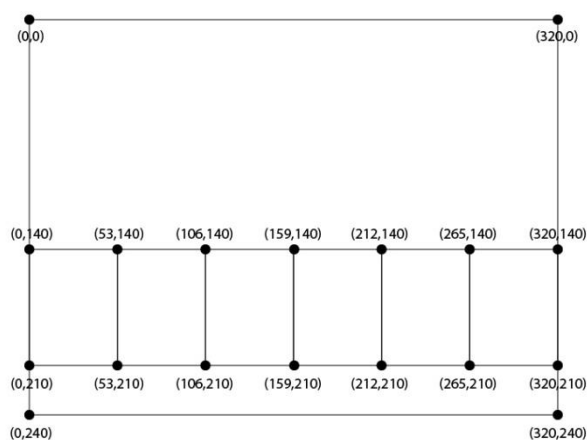
Bertujuan untuk mengetahui sistem dapat bekerja dengan cara memberi data input berupa video sehingga sistem dapat bekerja dengan semestinya ketika digunakan saat di jalan sebenarnya

6.1.2 Prosedur Pengujian Pendeteksi Kendaraan

Didalam pengujian keseluruhan sistem ini, peneliti menerapkan 6 kondisi pancaran cahaya yang berbeda dengan tujuan untuk mengetahui apakah sistim dapat menerima respon dari kondisi semua pancaran cahaya yang telah diterapkan.

6.1.3 Hasil dari Pengujian

Untuk mengetahui lokasi pancaran cahaya dibuat sebuah gambaran sederhana yang berguna untuk menampilkan titik kordinat garis. Dari gambaran tersebut dapat digunakan sebagai dasaran dalam mengetahui area pancaran cahaya tersebut. Gambaran tersebut dapat di liat pada gambar 6.13.



Gambar 6.1 Gambaran Sederhana Pembacaan Area Berdasar Titik Kordinat

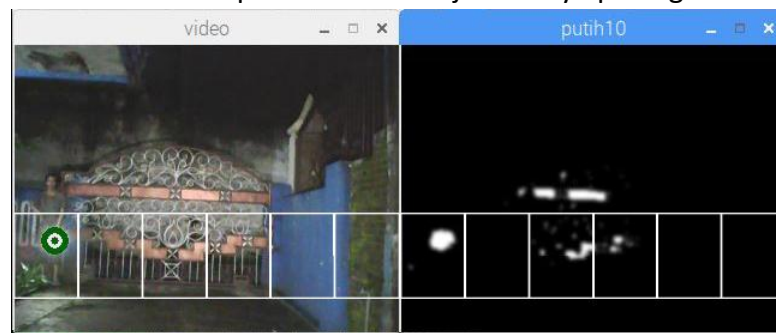
Berikut adalah hasil dari pengujian sistem secara keseluruhan dengan menampilkan hasil kamera dan hasil lampu.

1. Percobaan kondisi 1

Tabel 6.1Potongan Program Deteksi Pancaran Cahaya dengan Area yang telah ditentukan

126	<pre>if(KorXYRputih10[0]>=0 && KorXYRputih10[0]<=53 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){</pre>
-----	---------------------------------------------------------------------------------------------------------------------------------------------------

Dari potongan program tersebut diatas menjelaskan area yang telah ditentukan untuk mematikan sebuah *led* ketika area tersebut terdeteksi objek cahaya. Berikut adalah hasil pendeteksian objek cahaya pada gambar 6.2.



(a)

(b)

Gambar 6.2Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil *Thresholding*

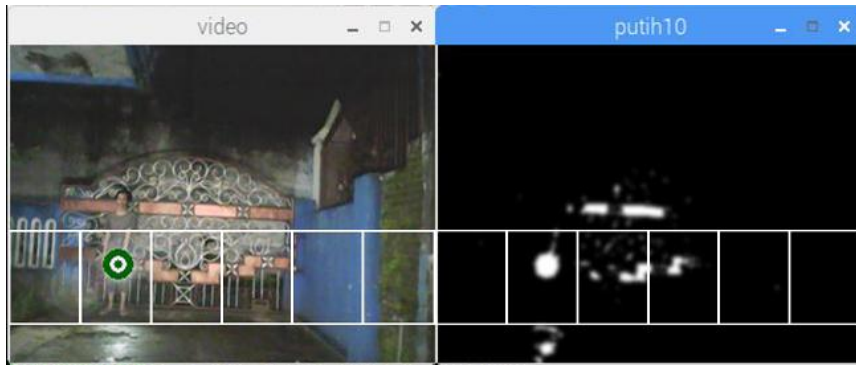
Pada bagian a menjelaskan seorang dengan memegang sebuah *handphone* dengan mengaktifkan lampu *flash* kemudian objek yang terdeteksi ditandai dengan lingkaran hijau, dan pada bagian b menjelaskan hasil threshold yang digunakan untuk acuan pendeteksian.

2. Percobaan kondisi 2

Tabel 6.2Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan

128	<pre>else if(KorXYRputih10[0]>53 && KorXYRputih10[0]<=106 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){</pre>
-----	---------------------------------------------------------------------------------------------------------------------------------------------------------

Dari potongan program tersebut diatas menjelaskan area yang telah ditentukan untuk mematikan sebuah *led* ketika area tersebut terdeteksi objek cahaya. Berikut adalah hasil pendeteksian objek cahaya pada gambar 6.3.



Gambar 6.3Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil *Thresholding*

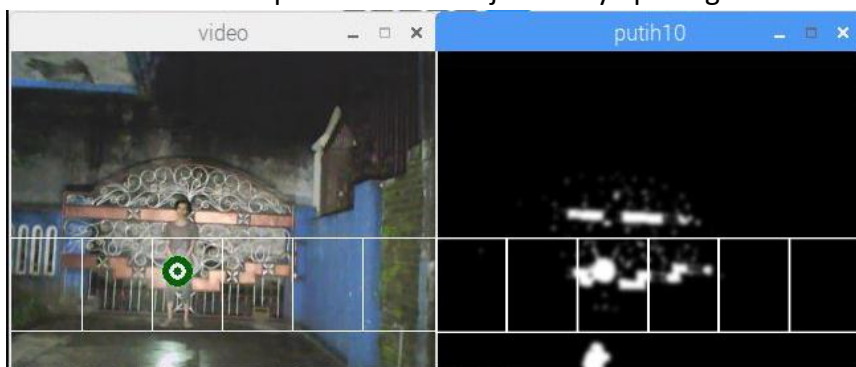
Pada bagian a menjelaskan seorang dengan memegang sebuah *handphone* dengan mengaktifkan lampu *flash* kemudian objek yang terdeteksi ditandai dengan lingkaran hijau, dan pada bagian b menjelaskan hasil threshold yang digunakan untuk acuan pendeteksian.

3. Percobaan Kondisi 3

Tabel 6.3Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan

130	<pre>else if(KorXYRputih10[0]>106 && KorXYRputih10[0]<=159 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210) {</pre>
-----	-----------------------------------------------------------------------------------------------------------------------------------------------------------

Dari potongan program tersebut diatas menjelaskan area yang telah ditentukan untuk mematikan sebuah *led* ketika area tersebut terdeteksi objek cahaya. Berikut adalah hasil pendeteksian objek cahaya pada gambar 6.4.



Gambar 6.4Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil *Thresholding*

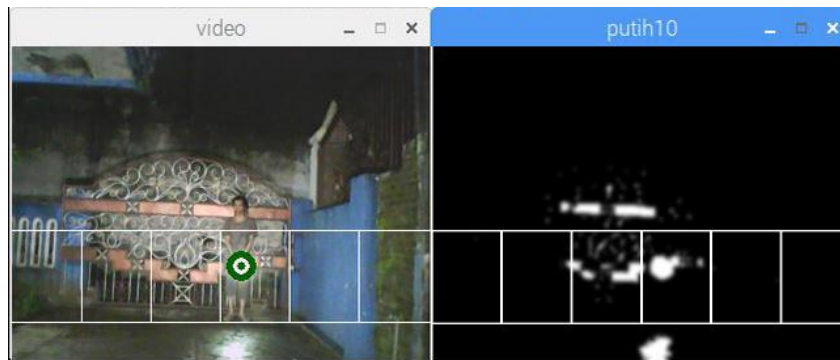
Pada bagian a menjelaskan seorang dengan memegang sebuah *handphone* dengan mengaktifkan lampu *flash* kemudian objek yang terdeteksi ditandai dengan lingkaran hijau, dan pada bagian b menjelaskan hasil threshold yang digunakan untuk acuan pendeteksian.

4. Percobaan Kondisi 4

Tabel 6.4 Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan

132	<pre>else if(KorXYRputih10[0]>159 && KorXYRputih10[0]<=212 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){</pre>
-----	----------------------------------------------------------------------------------------------------------------------------------------------------------

Dari potongan program tersebut diatas menjelaskan area yang telah ditentukan untuk mematikan sebuah *led* ketika area tersebut terdeteksi objek cahaya. Berikut adalah hasil pendeteksian objek cahaya pada gambar 6.5.



Gambar 6.5 Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil *Thresholding*

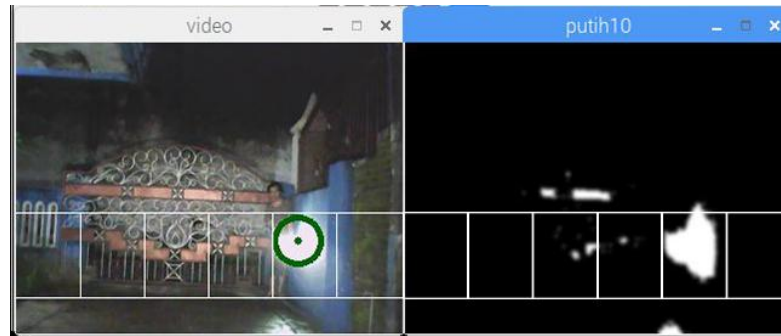
Pada bagian a menjelaskan seorang dengan memegang sebuah *handphone* dengan mengaktifkan lampu *flash* kemudian objek yang terdeteksi ditandai dengan lingkaran hijau, dan pada bagian b menjelaskan hasil threshold yang digunakan untuk acuan pendeteksian.

5. Percobaan Kondisi 5

Tabel 6.5 Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan

134	<pre>else if(KorXYRputih10[0]>212 && KorXYRputih10[0]<=265 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){</pre>
-----	----------------------------------------------------------------------------------------------------------------------------------------------------------

Dari potongan program tersebut diatas menjelaskan area yang telah ditentukan untuk mematikan sebuah *led* ketika area tersebut terdeteksi objek cahaya. Berikut adalah hasil pendeteksian objek cahaya pada gambar 6.20.



Gambar 6.6Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil *Thresholding*

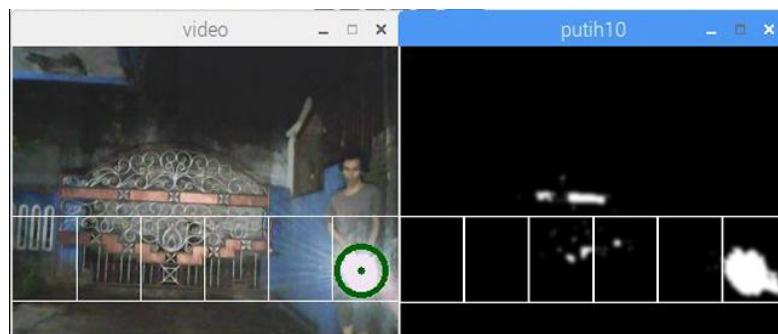
Pada bagian a menjelaskan seorang dengan memegang sebuah *handphone* dengan mengaktifkan lampu *flash* kemudian objek yang terdeteksi ditandai dengan lingkaran hijau, dan pada bagian b menjelaskan hasil threshold yang digunakan untuk acuan pendeteksian.

6. Percobaan Kondisi 6

Tabel 6.6Potongan Program Pendeteksian Pancaran Cahaya dengan Area yang telah ditentukan

136	<pre>else if(KorXYRputih10[0]>=265 && KorXYRputih10[0]<=320 && KorXYRputih10[1]>=140 && KorXYRputih10[1]<=210){</pre>
-----	-----------------------------------------------------------------------------------------------------------------------------------------------------------

Dari potongan program tersebut diatas menjelaskan area yang telah ditentukan untuk mematikan sebuah led ketika area tersebut terdeteksi objek cahaya. Berikut adalah hasil pendeteksian objek cahaya pada gambar 6.22.



Gambar 6.7Hasil Pendeteksi Area Pertama, (a) lingkaran pada gambar asli, (b) hasil *Thresholding*

Pada bagian a menjelaskan seorang dengan memegang sebuah *handphone* dengan mengaktifkan lampu *flash* kemudian objek yang terdeteksi ditandai dengan lingkaran hijau, dan pada bagian b menjelaskan hasil threshold yang digunakan untuk acuan pendeteksian.

6.1.4 Analisis Pengujian

Berdasarkan hasil pengujian menunjukkan bahwa dari 6 kondisi pancaran cahaya yang diujikan, untuk pengujian pendeteksi pancaran cahaya cukup baik karena walaupun hasil threshold menunjukkan objek terdeteksi cukup banyak, tetapi objek yang dilingkari hanyalah objek yang memiliki sebuah pancaran cahaya.

6.2 Pengujian Pengontrolan Lampu

6.2.1 Tujuan

Pengontrolan lampu yang akan di gunakan yaitu sejumlah 12 *led* yang akan dikontrol secara individu untuk dapat mengetahui apakah sistem dapat bekerja dengan maksimal.

6.2.2 Prosedur Pengujian Pendeteksi Kendaraan

Pengujian ini dilakukan dengan cara melakukan percobaan sebanyak 5 kali dan dilakukan dengan memberi *inputled* yang bervariasi. Kemudian di hitung apakah ada *led* bekerja dengan semestinya atau tidak.

6.2.3 Hasil dari Pengujian

Berikut adalah proses sebelum melakukan proses pengujian *led*, pertama yang dilakukan adalah menginisialisai pin yang digunakan terdapat pada tabel 6.6.

Tabel 6.7 Potongan Program Inisialisasi Pin GPIO Raspberry Pi

1	<code>int led1 = 29;</code>
2	<code>int led2 = 25;</code>
3	<code>int led3 = 24;</code>
4	<code>int led4 = 23;</code>
5	<code>int led5 = 22;</code>
6	<code>int led6 = 21;</code>
7	<code>int led7 = 3;</code>
8	<code>int led8 = 2;</code>
9	<code>int led9 = 0;</code>
10	<code>int led10 = 7;</code>
11	<code>int led11 = 27;</code>
12	<code>int led12 = 28;</code>

Kemudian pin di atur mode dengan *modeoutput* untuk memberi *input* dari *RaspberryPi* dan *output* pada led itu sendiri terdapat pada tabel 6.7.

Tabel 6.8Potongan Program Pengaturan Mode Pin

21	<code>pinMode (led1, OUTPUT) ;</code>
22	<code>pinMode (led2, OUTPUT) ;</code>

23	<code>pinMode (led3, OUTPUT) ;</code>
24	<code>pinMode (led4, OUTPUT) ;</code>
25	<code>pinMode (led5, OUTPUT) ;</code>
26	<code>pinMode (led6, OUTPUT) ;</code>
27	<code>pinMode (led7, OUTPUT) ;</code>
28	<code>pinMode (led8, OUTPUT) ;</code>
29	<code>pinMode (led9, OUTPUT) ;</code>
30	<code>pinMode (led10, OUTPUT) ;</code>
31	<code>pinMode (led11, OUTPUT) ;</code>
32	<code>pinMode (led12, OUTPUT) ;</code>

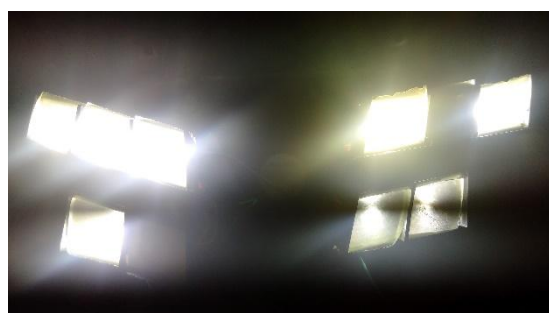
Dari hasil proses di atas, kemudian melakukan beberapa percobaan dengan membuat beberapa kondisi yang di atur secara bervariasi. Hasil percobaan dapat dilihat dibawah ini.

1. Percobaan 1

Tabel 6.9Potongan Program Percobaan Pengondisian *Led1*

36	<code>digitalWrite (led1, 0) ;</code>
37	<code>digitalWrite (led2, 1) ;</code>
38	<code>digitalWrite (led3, 0) ;</code>
39	<code>digitalWrite (led4, 1) ;</code>
40	<code>digitalWrite (led5, 1) ;</code>
41	<code>digitalWrite (led6, 0) ;</code>
42	<code>digitalWrite (led7, 1) ;</code>
43	<code>digitalWrite (led8, 1) ;</code>
44	<code>digitalWrite (led9, 1) ;</code>
45	<code>digitalWrite (led10, 1) ;</code>
46	<code>digitalWrite (led11,0) ;</code>
47	<code>digitalWrite (led12, 1) ;</code>

Hasil dari potongan program di atas akan menghasilkan pancaran cahaya seperti gambar 6.6.



Gambar 6.8Kondisi percobaan lampu 1

2. Percobaan 2

Tabel 6.10Potongan Program Percobaan Pengondisian *Led 2*

36	<code>digitalWrite (led1, 1) ;</code>
37	<code>digitalWrite (led2, 1) ;</code>
38	<code>digitalWrite (led3, 0) ;</code>
39	<code>digitalWrite (led4, 1) ;</code>
40	<code>digitalWrite (led5, 1) ;</code>
41	<code>digitalWrite (led6, 1) ;</code>
42	<code>digitalWrite (led7, 1) ;</code>
43	<code>digitalWrite (led8, 0) ;</code>
44	<code>digitalWrite (led9, 1) ;</code>
45	<code>digitalWrite (led10, 0) ;</code>
46	<code>digitalWrite (led11,1) ;</code>
47	<code>digitalWrite (led12, 1) ;</code>

Hasil dari potongan program di atas akan menghasilkan pancaran cahaya seperti gambar 6.7.



Gambar 6.9Kondisi percobaan lampu 2

3. Percobaan 3

Tabel 6.11Potongan Program Percobaan Pengondisian *Led3*

36	<code>digitalWrite (led1, 1) ;</code>
37	<code>digitalWrite (led2, 1) ;</code>
38	<code>digitalWrite (led3, 0) ;</code>
39	<code>digitalWrite (led4, 1) ;</code>
40	<code>digitalWrite (led5, 0) ;</code>
41	<code>digitalWrite (led6, 1) ;</code>
42	<code>digitalWrite (led7, 0) ;</code>
43	<code>digitalWrite (led8, 0) ;</code>
44	<code>digitalWrite (led9, 1) ;</code>
45	<code>digitalWrite (led10, 0) ;</code>
46	<code>digitalWrite (led11,1) ;</code>
47	<code>digitalWrite (led12, 0) ;</code>

Hasil dari potongan program di atas akan menghasilkan pancaran cahaya seperti gambar 6.8.



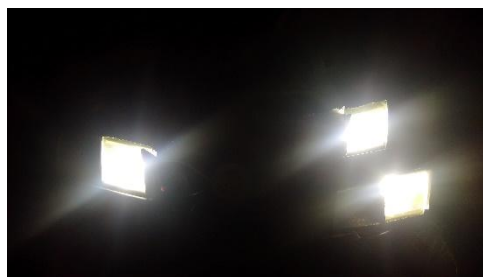
Gambar 6.10Kondisipercobaan lampu 3

4. Percobaan 4

Tabel 6.12Potongan Program Percobaan Pengondisian *Led4*

36	<code>digitalWrite (led1, 0) ;</code>
37	<code>digitalWrite (led2, 0) ;</code>
38	<code>digitalWrite (led3, 0) ;</code>
39	<code>digitalWrite (led4, 0) ;</code>
40	<code>digitalWrite (led5,0) ;</code>
41	<code>digitalWrite (led6, 1) ;</code>
42	<code>digitalWrite (led7, 0) ;</code>
43	<code>digitalWrite (led8, 0) ;</code>
44	<code>digitalWrite (led9, 1) ;</code>
45	<code>digitalWrite (led10, 0) ;</code>
46	<code>digitalWrite (led11,1) ;</code>
47	<code>digitalWrite (led12, 0) ;</code>

Hasil dari potongan program di atas akan menghasilkan pancaran cahaya seperti gambar 6.9.



Gambar 6.11Kondisi percobaan lampu 4

5. Percobaan 5

Tabel 6.13Potongan Program Percobaan Pengondisian *Led5*

36	<code>digitalWrite (led1, 0) ;</code>
37	<code>digitalWrite (led2, 0) ;</code>
38	<code>digitalWrite (led3, 0) ;</code>

```

39 digitalWrite (led4, 1) ;
40 digitalWrite (led5, 1) ;
41 digitalWrite (led6, 1) ;
42 digitalWrite (led7, 1) ;
43 digitalWrite (led8, 1) ;
44 digitalWrite (led9, 1) ;
45 digitalWrite (led10, 0) ;
46 digitalWrite (led11, 0) ;
47 digitalWrite (led12, 0) ;

```

Hasil dari potongan program di atas akan menghasilkan pancaran cahaya seperti gambar 6.10.



Gambar 6.12 Kondisi percobaan lampu 5

6.2.4 Analisis Pengujian

Berdasarkan dari hasil pengujian menunjukkan bahwa lampu yang dikontrol dapat bekerja dengan data input yang telah dimasukkan sebelumnya. Sehingga untuk mengontrol lampu sendiri tidak ada kendala dalam mengontrol secara individual. Dengan hasil tersebut maka sistem led yang dikerjakan secara individu akan digunakan untuk mengujian selanjutnya.

6.3 Pengujian Validitas Sistem

6.3.1 Tujuan

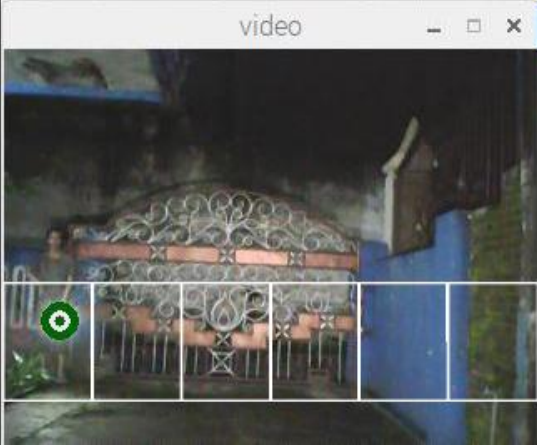

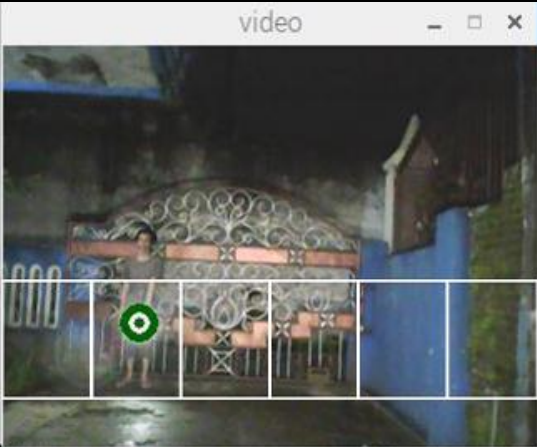

Bertujuan untuk mengetahui validitas dari pendeteksian kendaraan dan pengontrolan lampu *led matrix*.

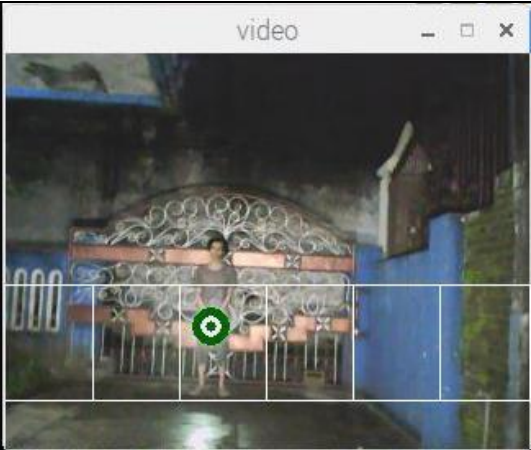
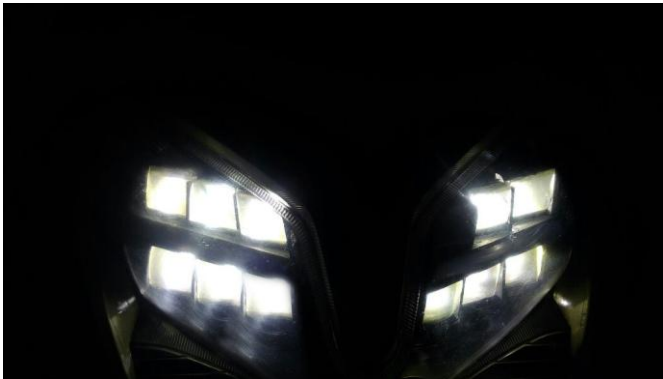
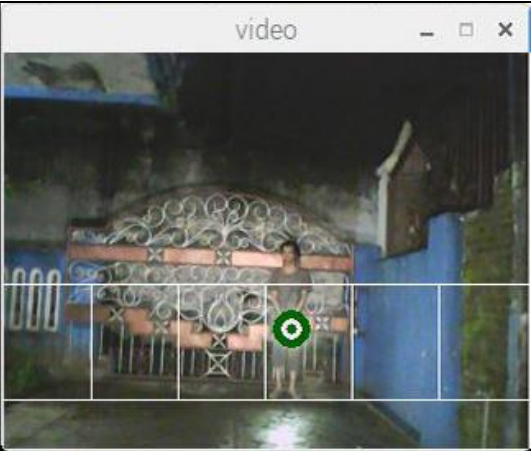

6.3.2 Prosedur Pengujian Validitas Sistem

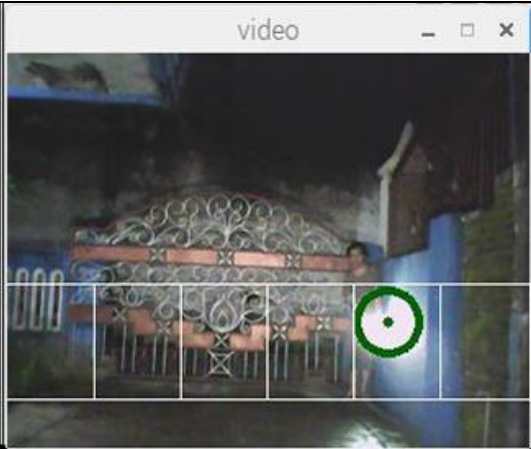
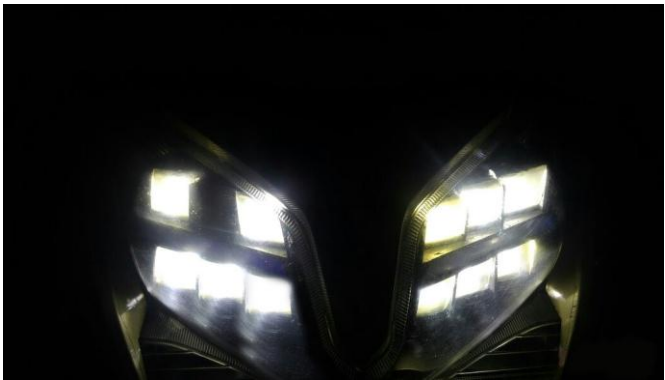
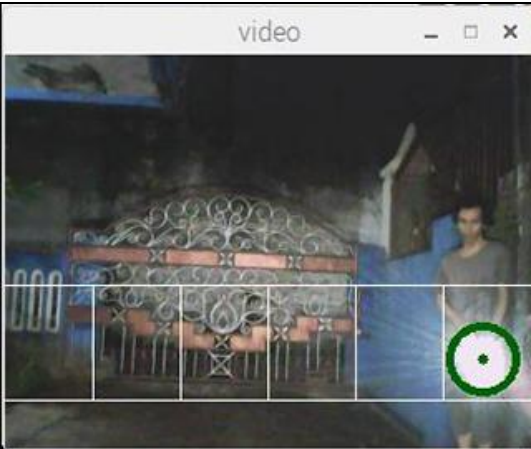
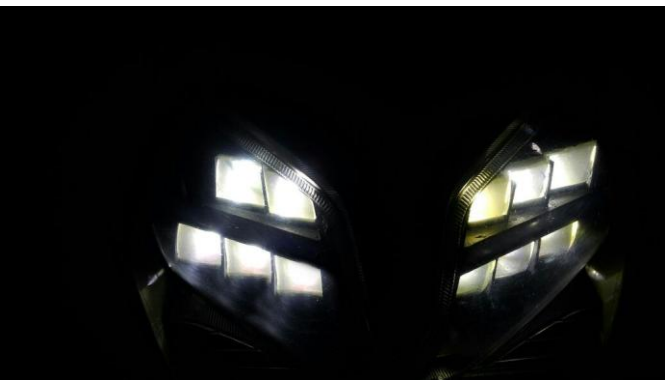
Didalam pengujian keseluruhan sistem ini peneliti menerapkan 6 kondisi pancaran cahaya yang berbeda, lalu sistem dijalankan dan di cek apakah hasil dari pengambilan video dan lampu led matrix memiliki kondisi yang sama.

6.3.3 Hasil dari Pengujian

Tabel 6.14 Validitasi Sistem Pendeteksi Objek dan Pengontrol Lampu

Kondisi	Hasil endeteksi video		Hasil pengontrolan lampu	Validitas
1				Valid
2				Valid

3	 A video player window titled 'video' showing a person standing in a doorway with a decorative archway. A green target icon is overlaid on the person.	 A close-up image of a car's headlights, which are illuminated, set against a dark background.	Valid
4	 A video player window titled 'video' showing a person standing in a doorway with a decorative archway. A green target icon is overlaid on the person.	 A close-up image of a car's headlights, which are illuminated, set against a dark background.	Valid

5			Valid
6			Valid

6.3.4 Analisis Pengujian

Berdasarkan hasil pengujian menunjukkan bahwa dari 6 kondisi pancaran cahaya yang diujikan, memiliki hasil validitasi yaitu sebesar 100% dimana hasil dari pendeteksian objek dapat terdeteksi dan pengontrolan lampu bekerja dengan mengikuti hasil dari pendeteksian objek tersebut.

PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem ini dalam mendeteksi kendaraan dengan menggunakan kamera dapat digunakan menggunakan metode *hough circle transform*. Penggunaan metode ini dapat mendeteksi kendaraan dengan mengambil pancaran cahaya dengan baik. Hal tersebut dapat diketahui dari hasil pengambilan gambar yang diterima dari *webcamLogitech C170*.
2. Sistem ini mampu mengontrol beberapa *led* yang disusun secara paralel dengan menggunakan komponen *transistor* BD139, komponen tersebut dapat digunakan sebagai saklar dalam mengontrol sejumlah *led* yang telah ditentukan oleh sistem.
3. Sistem ini memiliki hasil validitasi 100%, karena pendeteksian objek dan pengontrolan lampu dapat bekerja dengan sesuai dengan kondisi yang telah dibuat sebelumnya.

7.2 Saran

Beberapa saran yang dapat dikembangkan dalam sistem ini agar dapat menghasilkan sistem yang lebih baik, yaitu :

1. pengembangan selanjutnya di sarankan menggunakan mikrokomputer yang lain dengan kecepatan proses yang lebih cepat.
2. Pada pengembangan selanjutnya di sarankan menggunakan module kamera dengan resolusi kamera lebih besar.
3. Pada pengembangan selanjutnya di sarankan menggunakan desain yang dapat digunakan diberbagai jenis kendaraan.

DAFTAR PUSTAKA

- Azmi, F., 2006. *Embedded System : Sistem Kecil Nan Handal Yang Jarang Dikenal*. Sekolah Tinggi Teknologi Telkom Bandung.
- Bradski dan Kaehler, 2008, *Learning OpenCV, O'Really Publication*. New York.
- Fallah, Alborz. 2013. *Audi Matrix LED headlights technology explained*, (Online), (<http://www.caradvice.com.au/257871/audi-matrix-led-headlights/>. Diakses 5 februari 2017)
- Feng LUO & Fengjian HU. 2014. *A Comprehensive Survey Of Vision Based Vehicle Intelligent Front Light System*, (Online), (<http://www.s2is.org/issues/v7/n2/papers/paper14.pdf>. Diakses 1 februari 2016)
- Hu, T.H., 2012. Introduction to Digital Image Processing, Dept. of Electrical and Computer Engineering, ECE533 Digital Image Processing Univ of Wisconsin.
- Huaman, A., 2012. OpenCV Reference Manual, (Online), (www.opencv.org/download/manual/OpenCV_Manual.pdf), diakses pada 7februari 2017)
- Ipanase. 2014. Lampu HID silau dan berbahaya, kenali hukumnya (Online), (<http://pertamax7.com/2014/01/02/lampu-hid-silau-dan-berbahaya-kenali-hukumnya/>. Diakses 1 februari 2016)
- Modifikasi, 2014. Perbedaan Lampu Headlight Menggunakan Halogen, Led, Hid dan Laser. (Online). (<http://www.modifikasi.com/showthread.php/531912-Perbedaan-Lampu-Headlight-Menggunakan-Halogen-Led-Hid-dan-Laser>. Diakses 8 februari 2017)
- Munir, R., 2004. *Kontur dan Representasinya*. Dept. Sekolah Tinggi Elektro dan Informatika, Institut Teknologi Bandung. (Online), (http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Pengolahan%20Citra%20Digital/Bab-9_Kontur%20dan%20Representasinya.pdf. Diakses pada 7 februari 2017)
- Obengplus, 2015. Menaikan tegangan atau menurunkan tegangan DC to Dc dengan LM2596 dan CN6009 untuk DC to DC. (Online). (<http://obengplus.com/artikel/articles/226/1/Menaikan-tegangan-atau-menurunkan-tegangan-DC-to-DC-dengan-LM2596-dan-CN6009-untuk--DC-to-DC.html>. Diakses 8 februari 2017)
- OpenCV. 2012. OpenCV Documents. OpenCV It See, (Online), (<http://docs.opencv.org/doc/>, diakses tanggal 5 februari 2017)
- Putra, D., 2010. Pengolahan Citra Digital. Yogyakarta: Andi.

- Rahmatullah, Alki. 2016. Lampu led putih masuk surat kabar karena terlalu sialu setujukah, (Online), (<https://indoride.com/2016/01/20/lampu-led-putih-masuk-surat-kabar-karena-terlalu-sialu-setujukah/>). Diakses 25 Januari 2016)
- Spinks, Jez. 2016. Audi Matrix LED Headlight Technology: does it work?, (Online), (<https://www.whichcar.com.au/car-advice/audi-matrix-led-headlight-technology-does-it-work>). Diakses 5 februari 2017)
- Suranata, Aditya. Raspberry pi 3 Telah Dirilis! Berikut Spesifikasi, Uji Performa & Lainnya, (Online),(<https://tutorkeren.com/artikel/raspberry-pi-3-telah-dirilis-berikut-spesifikasi-uji-performa-lainnya.htm> ,diakses pada 7 februari 2017)
- Tamburo, Robert. 2014. *Programmable Automotive Headlights*, (Online), (<http://www.utc.ices.cmu.edu/utc/TNCCRKN-ECCV14.pdf>. Diakses 1 februari 2016)
- Vujovic, V., Maksimovic, M., 2014. *Raspberry Pi as a Wireless Sensor Node: Performance and Constraints*. ISBN, 978-953-233-081-6 [e-journal]. (Online).(<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6859717>. Diakses pada 7februari 2017)
- Wibowo, S., 2010. *Web Camera*. (Online).(http://satrio1505.blogspot.co.id/2013/03/webcam_3685.html. Diakses 7Februari 2017)

LAMPIRAN

L 1Lampiran Source Code

```
1 //memasukkan library opencv
2 #include<opencv2/core/core.hpp>
3 #include<opencv2/highgui/highgui.hpp>
4 #include<opencv2/imgproc/imgproc.hpp>
5 #include<opencv/cvaux.h>
6 #include<opencv/highgui.h>
7 #include<opencv/cxcore.h>
8
9 #include<iostream>
10 #include <stdio.h>
11 #include <stdlib.h>
12
13 //memasukkan library gpio
14 #include <wiringPi.h>
15
16 //memasukkan perhitungan
17 #include <vector>
18 #include <cmath>
19 #include <numeric>
20
21 using namespace std;
22 using namespace cv;
23
24 int i;//inisialisai untuk perulangan
25
26 //digunakan untuk memilih warna yang akan digunakan
27 IplImage *GetThresholdedImage(IplImage *HSV_putih)
28 {
29     IplImage *putih10 =
30     cvCreateImage(cvGetSize(HSV_putih),IPL_DEPTH_8U, 1);//membuat
    variabel gambar baru
31     cvInRangeS(HSV_putih,//masukkan frame
32     cvScalar(0,0,200,0),//nilai minimum HSV
33     cvScalar(180,255,255,0),//nilai maksimum HSV
34     putih10);//keluaran frame
35     return putih10;
36 }
37
38 int main(int argc, char* argv[]) {
39 //inisialisai pin gpio yang akan digunakan
40     system("gpio export 21 out");
41     system("gpio export 26 out");
42     system("gpio export 19 out");
43     system("gpio export 13 out");
44     system("gpio export 6 out");
45     system("gpio export 5 out");
46     system("gpio export 22 out");
47     system("gpio export 27 out");
48     system("gpio export 17 out");
49     system("gpio export 4 out");
50     system("gpio export 16 out");
51     system("gpio export 20 out");
52     system("gpio export 18 in");
53
54     wiringPiSetupSys(); //memanggil pin gpio tanpa root
55
56     bool quit = false; //digunakan untuk mengakhiri program
57     CvCapture* video;//inisialisasi video
58     video = cvCaptureFromCAM(0);//digunakan untuk mengaktifkan
59     kamera
```

```

60 //video = cvCaptureFromFile("video/3.mp4");//digunakan
61 untuk mengambil video
62 cvGrabFrame(video);//mengambil perframe pada hasil kamera
63 //IplImage* gambar_asli = cvQueryFrame(video);//mengambil
64 perframe pada video
65 IplImage * gambar_asli =
66 cvRetrieveFrame(video);//mengambil hasil frame dengan nama
67 variabel gambar_asli
68
69 //pengondisian jika video dapat dibaca
70 if (!gambar_asli) {
71     cout << "Error: cap Webcam not accessed
72 successfully\n\n";
73     return(0);
74 }
75 //CvHoughCircles membutuhkan memory storage untuk
76 penyimpanan memori saat pemrosesan
77 CvMemStorage* simpan_putih10; // deklarasi untuk memori
78 proses lampu
79
80 //CVHoughCircles membutuhkan pointer circles, untuk
81 deklarasi sebelum fungsi dieksekusi dengan fungsi
82 cvGetSeqElem(p_seqCircles,1) yang akan menghasilkan variabel
83 // koordinat x titik tengah, koordinat y titik tengah ,
84 dan jari-jari lingkaran
85 CvSeq* p_seqCirclesputih10; // pointer untuk deklarasi
86 sebelum dieksekusi
87
88 float* KorXYRputih10; // pointer untuk 3 elemen objek
89 terdeteksi;
90
91 cvNamedWindow("video", CV_WINDOW_AUTOSIZE); //pembuatan
92 window dengan nama video
93 cvNamedWindow("putih10", CV_WINDOW_AUTOSIZE); //pembuatan
94 window dengan nama putih10
95
96 while (!quit && cvGrabFrame(video)) //perulangan dengan
97 kondisi tidak keluar dan ada video
98 {
99     gambar_asli = cvRetrieveFrame(video); //mengambil
100 perframe pada hasil kamera
101 IplImage *ukuran_gambar = cvCreateImage(cvSize(320,
102 240),8,3); //inisialisasi frame dengan resolusi 320x240
103
104 cvResize(gambar_asli,ukuran_gambar,CV_INTER_LINEAR);//memp
105 erkecil frame
106 IplImage *HSV_putih =
107 cvCreateImage(cvGetSize(ukuran_gambar),IPL_DEPTH_8U,
108 3);//inisialisasi variabel HSV_putih
109 cvCvtColor(ukuran_gambar,//masukan frame
110 HSV_putih, //keluaran frame
111 CV_BGR2HSV);//algoritma perubahan warna dari RGB ke
112 HSV
113 IplImage *putih10 = GetThresholdedImage(HSV_putih);
114 //pengondisian warna terdeteksi
115
116 simpan_putih10 = cvCreateMemStorage(0); //menyimpan
117 memori hasil proses
118
119 CV_GAUSSIAN, //menggunakan gaussian blur
120 9, //tinggi kernel
121 9); // lebar kernel

```

```

131
132         p_seqCirclesputih10 = cvHoughCircles(putih10, //
masukan frame
133             simpan_putih10, //keluaran frame
134             CV_HOUGH_GRADIENT, // algoritma berbasis
gradien warna
135             2, // resolusi gradient pixel
136             40, // minimal jarak
137             255, // batas atas threshold
138             40, // batas bawah threshold
139             8, // min jari-jari lingkaran dalam pixel
140             40); // max jari-jari lingkaran dalam pixel
141
142
143         for(i=0; i <p_seqCirclesputih10->total;
i++){KorXYRputih10 = (float*)cvGetSeqElem(p_seqCirclesputih10,
i); //perulangan dengan kordinat x, y dan area
144
145             cvCircle(ukuran_gambar, // gambar di frame
ROI
156             cvPoint(cvRound(KorXYRputih10[0]),
157             cvRound(KorXYRputih10[1])), //tengah
centroid
158             3, // 3 pixel jari lingkarn
159             CV_RGB(0,100,0), // warna hijau
160             CV_FILLED); //diisi dengan ketebalan 3 pixel
161
162             //pembuatan lingkaran dengan kordinat x dan
y
163
164             cvCircle(ukuran_gambar,cvPoint(cvRound(KorXYRputih10[0]),
165             cvRound(KorXYRputih10[1])),cvRound(KorXYRputih10[2]),CV_RGB
B(0,100,0),3);
166
167             //Buat garis bantu
168             //garis vertikal
169             cvLine(ukuran_gambar, Point(0, 140),
170             Point(320, 140), Scalar(255, 255, 255), 1, 8);
171             cvLine(ukuran_gambar, Point(0, 210),
172             Point(320, 210), Scalar(255, 255, 255), 1, 8);
173
174             //garis horizontal
175             cvLine(ukuran_gambar, Point(53, 140),
176             Point(53, 210), Scalar(255, 255, 255), 1, 8);
177             cvLine(ukuran_gambar, Point(106, 140),
178             Point(106, 210), Scalar(255, 255, 255), 1, 8);
179             cvLine(ukuran_gambar, Point(159, 140),
180             Point(159, 210), Scalar(255, 255, 255), 1, 8);
181             cvLine(ukuran_gambar, Point(212, 140),
182             Point(212, 210), Scalar(255, 255, 255), 1, 8);
183             cvLine(ukuran_gambar, Point(265, 140),
184             Point(265, 210), Scalar(255, 255, 255), 1, 8);
185
186             //pengondisian jika sistem di aktifkan
187             if (digitalRead(18)==1)
188             {
189                 //pengondisian sistem aktif atau tidak
190                 //pengondisian lampu bawah
aktif
191                 digitalWrite (21, HIGH) ;
192                 digitalWrite (26, HIGH) ;
193                 digitalWrite (19, HIGH) ;
194                 digitalWrite (13, HIGH) ;
195                 digitalWrite (6, HIGH) ;

```

```

190 digitalWrite (5, HIGH) ;
191
192 //pengondisian per lampu dengan
area yang telah ditentukan
193 if(KorXYRputih10[0]>=0      &&
KorXYRputih10[0]<=53      &&      KorXYRputih10[1]>=140      &&
KorXYRputih10[1]<=210) {
194     digitalWrite (4, LOW) ;}
195     else if(KorXYRputih10[0]>53 &&
KorXYRputih10[0]<=106      &&      KorXYRputih10[1]>=140      &&
KorXYRputih10[1]<=210) {
196     digitalWrite (16, LOW) ;}
197     else if(KorXYRputih10[0]>106 &&
KorXYRputih10[0]<=159      &&      KorXYRputih10[1]>=140      &&
KorXYRputih10[1]<=210) {
198     digitalWrite (20, LOW) ;}
199     else if(KorXYRputih10[0]>159 &&
KorXYRputih10[0]<=212      &&      KorXYRputih10[1]>=140      &&
KorXYRputih10[1]<=210) {
200     digitalWrite (22, LOW) ;}
201     else if(KorXYRputih10[0]>212 &&
KorXYRputih10[0]<=265      &&      KorXYRputih10[1]>=140      &&
KorXYRputih10[1]<=210) {
202     digitalWrite (27, LOW) ;}
203     else if(KorXYRputih10[0]>=265 &&
KorXYRputih10[0]<=320      &&      KorXYRputih10[1]>=140      &&
KorXYRputih10[1]<=210) {
304     digitalWrite (17,
LOW) ;}
305
306     else{
307
308     digitalWrite (22, HIGH) ;
309     digitalWrite (27, HIGH) ;
310     digitalWrite (17, HIGH) ;
311     digitalWrite (4, HIGH) ;
312     digitalWrite (16, HIGH) ;
313     digitalWrite (20, HIGH) ;
314     }
315     //pengondisian jika sistem mati maka
lampu hidup semua
316     }else{
317     digitalWrite (21, HIGH) ;
318     digitalWrite (26, HIGH) ;
319     digitalWrite (19, HIGH) ;
320     digitalWrite (13, HIGH) ;
321     digitalWrite (6, HIGH) ;
322     digitalWrite (5, HIGH) ;
323     digitalWrite (22, HIGH) ;
324     digitalWrite (27, HIGH) ;
325     digitalWrite (17, HIGH) ;
326     digitalWrite (4, HIGH) ;
327     digitalWrite (16, HIGH) ;
328     digitalWrite (20, HIGH) ;
329     }
330 }
331
332
333
334 //menampilkan hasil kamera dengan hasil
pendeteksian dan garis bantu
335 cvShowImage("video", ukuran_gambar);
336 //menampilkan hasil pendeteksian dengan warna hitam
dan putih dimana putih adalah objek yang dideteksi
337 cvShowImage("putih10", putih10);

```

```
338
339         //pengondisian ketika mengakiri program
340         if (waitKey(30) == 27)
341         {
342             cout << "esc key is pressed by user" <<
endl;
343             quit = true;
344             break;
345         }
346     }
347     return 0;
348 }
349
350
```